Operator's Manual

W A V E F O R M - C A T A L Y S T

MULTICHANNEL DIGITAL OSCILLOSCOPE
AND INSTRUMENT CONTROL SYSTEM
USER'S MANUAL

Version 2.3

LeCroy

Innovators in Instrumentation

W A V E F O R M - C A T A L Y S T

MULTICHANNEL DIGITAL OSCILLOSCOPE
AND INSTRUMENT CONTROL SYSTEM
USER'S MANUAL

Version 2.3

Revised
August 1986

WARRANTY STATEMENT

LeCroy Research Systems Corporation
Software License Agreement

This license allows the use of this product on a single machine. Under this license you may:

1. Copy the software for backup or modification purposes in support of your use of the software on the single machine.

2. Modify the software and/or merge it into another program for your use on the single machine.

3. Transfer the software and the license to another party, if and only if, the other party accepts the terms of this agreement, and you relinquish all copies whether in printed or machine readable form including all modified or merged versions.

Limited Warranty

The software is provided "AS IS" without warranty of any kind, either expressed or implied, with respect to its performance, merchantability, or fitness for any particular purpose. The software is thoroughly tested and thought to be functional when released. At LeCroy's sole discretion, LeCroy may provide an update to correct any serious problems discovered in the software.

LeCroy warrants the diskette(s) on which the software is furnished to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery.

LeCroy Research Systems shall have no liability or responsibility to the purchaser or any other person or entity with respect to any liability, loss, or damage caused, or alleged to be caused, directly or indirectly by this software, including, but not limited to, any interruption of service, loss of business, or anticipatory profits, or consequential damages resulting from the use of this software.

Trademarks

IBM, IBM PC, and PC DOS, are registered trademarks of International Business Machines Corporation.

GPIB-PC is a trademark of National Instruments Corporation.

TABLE OF CONTENTS

		PAGE
1.	INTRODUCTION	1-1
2.	CONFIGURATION GUIDE	2-1
3.	SYSTEM ASSEMBLY	3-1
	3.1 Hardware Setup	
	3.2 Setting up CATALYST Disks	
4.	GETTING STARTED	4-1
5.	COMMAND DESCRIPTIONS	5-1
	5.1 Instrument Commands	5-1
	5.1.1 Setup Menu	5-2
	5.1.2 Edit Menu	5-4
	5.1.3 User Camac Menu	5-5
	5.2 Trigger Commands	5-7
	5.3 Display Controls	5-8
	5.4 Keypad Functions	5-12
	5.4.1 Trace Cursors	5-12
	5.4.2 Expand Trace	5-12
	5.4.3 Position Trace	5-12
	5.5 Averaging and Histogramming	5-14
	5.5.1 Averaging and Histogramming Menu	5-14
	5.5.2 Example of Averaging	5-17
	5.5.3 Example of Histogramming	5-17
	5.6 Autosequence	5-18
	5.6.1 Autosequence Menu	5-18
	5.6.2 Autosequence Learn Menu	5-19
	5.6.3 Autosequence Edit Menu	5-24
	5.6.4 Examples of Autosequence	5-26
	5.6.5 Notes About Using Autosequence	5-28

5.7 Hardcopy Options 5-30

5.8 Disk Utilities 5-32

5.9 Miscellaneous Controls 5-33

5.10 Waveform Limit Testing 5-34

6. CHANGING PROGRAM DEFAULT CONDITIONS 6-1

Appendix A: Format of Stored Data
Appendix B: Interface to Signal Processing Software
Appendix C: Function Generator Utility
Appendix D: Sinusoidal Fitting
Appendix E: Diagnostics

WAVEFORM-CATALYST™
DIGITAL STORAGE OSCILLOSCOPE

LeCroy

6900 DIGITAL STORAGE OSCILLOSCOPE SOFTWARE

- **Turnkey Waveform Recording**
- **Flexible Modular Design**
- **Up to 1.3 GS/sec Digitizing Rate**
- **32K to 16M Sample Records**
- **Signal Processing**
- **Fast Signal Averaging**

CONTROL, DISPLAY, ARCHIVE, AND PROCESS DATA FROM LeCROY MODULAR WAVEFORM DIGITIZERS

WAVEFORM-CATALYST™ Digital Storage Oscilloscope Software operating on an IBM® Personal Computer* provides a turnkey system for acquisition, display, storage, and processing of data obtained from LeCroy Modular Waveform Digitizers. The full line of LeCroy Modular Recorders, encompassing sampling rates from 500 samples/sec at 12-bits resolution to 1.3 gigasamples/sec at 8-bits, and with memories to 16 megasamples per channel, are supported by the software. In addition, LeCroy signal conditioning modules, trigger generators, clock generators, and arbitrary function generators are programmable from WAVEFORM-CATALYST, providing a complete measurement system for benchtop, laboratory or automatic test.

*IBM-PC, XT, or AT (or strict compatibles at user's responsibility).

FEATURES

Flexible Modular Recording—WAVEFORM-CATALYST operates with LeCroy Waveform Digitizer Modules; sampling rates 500 samples/sec to 1.3 Gigasamples/sec; memories expandable to 16M samples/channel.

Programmable—WAVEFORM-CATALYST provides programmable sampling speed, memory length, pretrigger samples, full page setup menu.

GPIB Compatible—Waveform Digitizers act as GPIB devices; up to 15 instrument mainframes—over 100 channels—are controlled by one IBM-PC.

Versatile Triggering—Single keystrokes of IBM-PC keyboard allow for autotrigger, external or single-shot triggering of Waveform Digitizers.

Signal Conditioning—Programmable amplifiers provide conditioning of input before digitizing.

Signal Averaging—Average up to 8K samples divided among 4 channels using either LeCroy Model 6010 or Model 8901A GPIB controller.

Signal Processing—WAVEFORM-CATALYST digitized data is fully compatible with ILS-PC™ and ASYST™ signal processing packages for the IBM-PC.

User Programmable—User-written routines may be added with Model 6900 WAVEFORM-CATALYST source code.

Archiving—Floppy disk storage of digitized records; recall and display of reference trace.

Display—One to 4 channels can be displayed simultaneously on screen; vertical and horizontal manipulations available on all traces.

Cursors—Time and voltage cursors provide absolute and relative measurements on displayed data.

Autosequence Mode—Sequences of WAVEFORM-CATALYST commands can be concatenated and stored for later execution.

Supports Other LeCroy Modules—Function generators, time-to-digital converters, gate and delay generators and other selected modules can operate from WAVEFORM-CATALYST.

Self Documenting—WAVEFORM-CATALYST stores digitized data complete with recording parameters and user-entered comments.

FUNCTIONAL DESCRIPTION

Waveform Recording hardware systems can be configured using LeCroy waveform recording modules and accessories. The selection of a module or modules is typically based on the following needs:

- Bandwidth
- Vertical Bit Resolution
- Memory Length
- Signal Conditioning

After selecting a module or modules that best meet your measurement needs, the module(s) may be placed in a LeCroy Model 8013A benchtop or Model 1434A rackmount instrument mainframe. These mainframes provide power to the module(s) and GPIB communication via a Model 8901A or 6010 GPIB module. Each instrument mainframe is recognized by the IBM-PC and WAVEFORM-CATALYST software as a separate GPIB device.

Using the WAVEFORM-CATALYST software, Triggering Modes, Recording Parameters, Signal averaging and display manipulation can be easily controlled with single keystrokes of the IBM-PC keyboard. With WAVEFORM-CATALYST, engineering time can be devoted to measurements and interpretation rather than programming and software development.

WAVEFORM-CATALYST systems provide maximum flexibility to control LeCroy instruments operating from hertz to gigahertz with the same software. A full screen setup menu allows for programming the appropriate sampling rate, pre-trigger samples, memory length, and offset of each module.

Up to 4 channels of measured or archived data may be displayed simultaneously on the PC monitor. Individual channels may be expanded, positioned, and overlaid for visual comparison. Dual cursors allow for absolute and relative measurements on interesting waveform phenomena. Long data records may be paged at 8K sample blocks or data blocks may be joined to view the total recorded signal.

High-speed signal averaging is provided with the LeCroy Model 6010 controller module or WAVEFORM-CATALYST. For averaging high-repetition-rate events (captured with LeCroy's high speed digitizing modules) the Model 6010 replaces the Model 8901A as both a signal averaging and GPIB I/O module. Operation of WAVEFORM-CATALYST averaging mode is transparent to the user for either 6010 or 8901A modules.

Data acquisition and signal processing can be done with the same system—without programming—using signal processing packages designed for the IBM-PC. Both ILS-PC and ASYST software packages provide frequency, time and statistical analyses of stored digitized data from LeCroy recorders. In addition, processing programs written in FORTRAN by the user can be incorporated into the data acquisition cycle with the source code provided with the Model 6900 WAVEFORM-CATALYST.

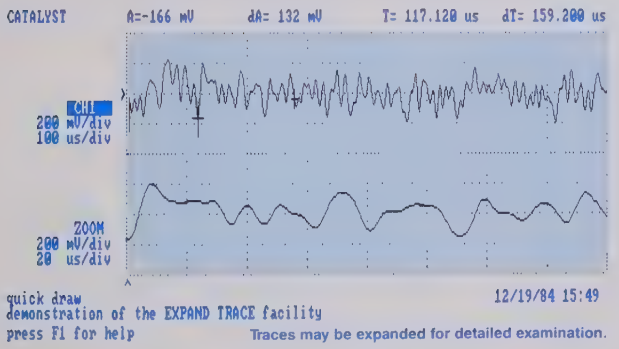
SPECIFICATIONS

MODULAR WAVEFORM DIGITIZER COMPATIBILITY
The 6900 is compatible with most modular LeCroy Waveform Digitizers. Sampling rates from 500 samples/sec to 1.3 gigasamples/sec; memories to 16M samples per channel; resolution, 8-12 bits. Consult module technical data sheets for appropriate digitizers and accessories.

OTHER LeCROY MODULES SUPPORTED
Programmable amplifiers, trigger generators, gate and delay generators, time-to-digital converters, arbitrary function generator, charge-integrating and voltage-sensing spectroscopy ADC's.

SYSTEM CAPACITY
Up to 15 instrument mainframes for each computer. Each mainframe acts as one GPIB device. Add power requirements and slots required for specific modules to determine maximum channels per mainframe.

WAVEFORM DISPLAY
Monochrome using CGA or Color (EGA) cards provide screen resolution 640 x 200; 512 H x 128 V for trace area; 1 to 4 channels can be displayed simultaneously; time and amplitude annotations provided; update rate to display over 70 times per minute on IBM-PC AT. Simple plotting and glitch-detecting algorithm provided for displaying data. Display 8K blocks of memory or join blocks to plot full memory on monitor. Hardcopies can be obtained with graphics printer (1 minute).



DISPLAY FEATURES
Horizontal and vertical expand in 1-2-5 sequences; no maximum on expansions; redefine screen origin to zoom in on waveform features; trace positioning horizontal and vertical; dual cursors provide absolute and relative measurements on waveforms to 1 μV and .1 nsec resolution; cursors annotated at top of screen; 160 characters of user annotation may be shown on each display.

INSTRUMENT CONTROL
Digitizer and conditioning parameters may be selected on PC in spreadsheet format; all operations performed by single keystrokes on PC keyboard.

TRIGGER MODES
Six trigger modes provided; Auto—computer-generated trigger after 3 seconds; Single—arms recorder and waits for external trigger; Normal—arms modules after every trigger; Manual—keyboard generated trigger; Freeze—inhibits incoming triggers; Average/Histogram—enters summation averaging, histogramming mode.

SIGNAL AVERAGING/HISTOGRAMMING
Average/Histogram 1 to 4 channels and store in 8K × 31-bit buffer memory; buffer divided between active channels; record length selectable from 64 to 8,192 samples; summation averaging rate on IBM-PC AT is 3-5 Hz with Model 8901A; 1-2 Hz rate using 6880A/6010 and 250 sample record; *high speed summation (130 Hz for 1024 samples) and exponential signal averaging using Model 6010 controller with Models TR8837F, TR8818, or TR8828C modular digitizers.* WAVEFORM-CATALYST operation is transparent on Models 8901A or 6010; histogramming mode displays counts vs. channel.

AUTOSEQUENCING
Sequence of commands can be learned for later execution; Interactive, Learn and Run modes define flow, assign variables and make *conditional decisions*; sequences store to disk and recall for later use; 25 variables available for computation; 4,000 lines of commands available for autosequence.

DATA STORAGE
Waveforms, instrument setup parameters, and user comments can be manually or automatically stored to disk; archiving rates are 1 kbyte/sec to floppy disk and 10 kbyte/sec to hard disk: data stored unformatted, variable length records, with comments; accessible from FORTRAN, BASIC or any programming language.

SIGNAL PROCESSING
Data files convert to format compatible with ILS-PC and ASYST signal processing packages (run off-line from WAVEFORM-CATALYST); user-written processing can be implemented with LeCroy 6900 WAVEFORM-CATALYST PLUS Source Code.

MODEL 6010 CONTROLLER
Acts both as smart GPIB module for instrument mainframe and as fast signal averager when used with LeCroy Models TR8828C, TR8818A, and TR8837F family of high speed digitizers. All CATALYST supported digitizers are compatible with the 6010 controller. Models 6010 and 8901A are transparent to user when operating CATALYST program.

CUSTOM PROCESSING CAPABILITY
Model 6900 WAVEFORM-CATALYST contains source code for all program subroutines; allows user to modify all acquisition, display, archiving, and processing routines using FORTRAN or assembly language; IBM FORTRAN Compiler Version 2.0 is required for modifications; not included in 6900.

WAVEFORM-CATALYST IBM-PC CONFIGURATION GUIDE
IBM-PC, XT, or AT, (or strict compatibles at user's responsibility).

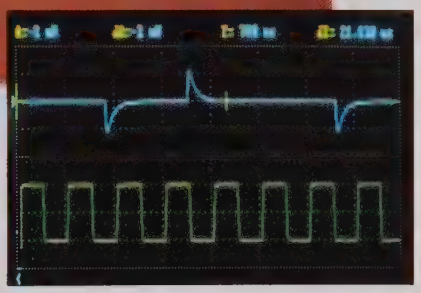
All Models Use:
IBM Color/Graphics Adapter card (CGA), IBM Enhanced Graphics Adapter (EGA) or strict equivalent.
National Instruments GPIB-PC2, PC2A or PC3 interface card.
Composite Video Monitor
At least one floppy disk drive.
PC-DOS 2.0 and above
LeCroy Model 8901A or 6010 GPIB interface
Instrument Mainframe (LeCroy 8013A or 1434A).

Additional Components and Exceptions:
PC or XT
Multifunction Memory Expansion Card 256K memory with Serial and Parallel Port, TECMAR P/N 24044 or strict equivalent.
On XT standard 10 megabyte hard disk replaces 1 floppy disk; 8087 math coprocessor.
AT
IBM "Enhanced AT" contains 30 megabyte hard disk, 1 floppy disk drive; 80287 math coprocessor

Tecmar, 6225 Cochran Road, Solon, OH. 216-348-0600
National Instruments, 12109 Technology Blvd, Austin, TX. 800-531-5066

ORDERING INFORMATION

6900	WAVEFORM-CATALYST Digital Storage Oscilloscope Software includes Executable Program, Source code, function library, National Instruments Software Handler, and Halo Graphics package.
6909-2	IBM PC FORTRAN 2.0
6900PCAT	IBM AT or COMPAQ 386 configured to run WAVEFORM-CATALYST software. Integrated and tested as system.
6900/386	Add C suffix for color display.



WAVEFORM-CATALYST

With LeCroy's high-performance modular waveform recorders, your data acquisition is turnkey for input signals from hertz to gigahertz. Oscilloscope-like display and control make the system easy to use from the moment power is applied. Spreadsheet menus provide simple programmability and recall of settings while optional modules provide high-speed averaging capability. Long recording memories, pioneered by LeCroy, are accommodated in the standard software package as are programmable preamplifiers, clock generators, and function generators.

LeCroy is the leader in high performance waveform digitizing. Modular recorders allow flexibility in sampling speed and memory length, all in the same mainframe. If you are challenged by a data acquisition problem and want the best in analog-to-digital recording, LeCroy has the solution.

ASYST is a registered trademark of MacMillan Software Co. (800) 348-0033
Halo Graphics is a trademark of Lifeboat Associates (212-860-0300)
IBM-PC is a registered trademark of International Business Machines Corporation
ILS-PC is a registered trademark of Signal Technology Inc. (800) 235-5787
WAVEFORM-CATALYST is a trademark of the LeCroy Corporation
COMPAQ® is a registered trademark of COMPAQ Computer Corporation

Copyright © May, 1987. LeCroy is a registered trademark of LeCroy Corporation. All rights reserved. Information in this publication supersedes all earlier versions. Specifications and prices subject to change without notice.

LeCroy

Innovators in Instrumentation

Corporate Headquarters
700 Chestnut Ridge Road
Chestnut Ridge, NY 10977-6499, USA
Tel: (914) 578-6038
TWX: 710-577-2832

European Headquarters
Route du Nant-d'Avril 101
1217 Meyrin 1-Geneva, Switzerland
Tel: (022) 82 33 55, Telex: 28230

Other sales and service representatives throughout the world.

1. INTRODUCTION

WAVEFORM CATALYST is a modular instrumentation control system that runs on an IBM PC or compatible personal computer. Its purpose is to allow you to start using your modules almost immediately by making it simple for you to control your instruments. Single keystroke commands make the system easy to learn and easy to use. No prior knowledge of computers, programming languages, or the industry standards, is needed.

CATALYST is a total system. It includes functions that allow you to program the instruments, acquire data from them, display the information, process, and archive it. Built-in customization options allow you to set up the system the way that you like. A system library is available which will allow you to create your own program. If you have CATALYST PLUS see the CATALYST Function Library Manual for complete details.

2. CONFIGURATION GUIDE

The following items are necessary to use **CATALYST:**

Personal Computer - IBM or compatible with:
- At least one disk drive (two are recommended)
- 512K RAM memory
- IBM color/graphics board or IBM Enhanced Graphics Adapter
- Video Monitor
- Intel Math coprocessor
- PC-DOS 2.x, 3.0 or 3.1
- National Instruments GPIB-PC Interface

LeCroy Model 8901A GPIB Controller, Model 6010 MAGIC Controller, or Model 9400 Digital Storage Oscilloscope.

LeCroy Model 8013A or Model 1434A Crates (Instrument Housings) or equivalent.

LeCroy CATALYST program disks.

What Boards **DO** and **DO NOT** work based on factory experience.

MULTIFUNCTION BOARDS FOR NON-AT'S

WORK	**DO NOT WORK**
Tecmar Wave and Bosun	
Tecmar Captain	
Quadram Quadboard	

GRAPHICS BOARDS

WORK	**DO NOT WORK**
IBM Color/Graphics Adapter	Hercules Graphics Card
Hercules Color Card	IBM Monochrome Adapter
Tecmar Graphics Tender	Tecmar Graphics Master
**IBM Enhanced Color Graphics	
**Vega Video 7	
**Quadram Quadega +	

** Color requires RGB monitor

GPIB

WORK	**DO NOT WORK**
National Instruments	Tecmar
models GPIB-PC2, PC2A, PC3	Ziatec
IBM GPIB P/N6451503	CEC
	Metrabyte

COMPUTERS

WORK	DO NOT WORK
IBM PPC, PC, XT, AT	COMPAQ Deskpro
COMPAQ Portable	
COMPAQ Portable 286	
Zenith 150, 160	
Olivetti M24	
AT&T 6300	

PRINTERS

WORK	DO NOT WORK
HP Thinkjet 2225C	
Epson FX-80 RX-80	
Quadram Quadjet	
IBM Graphics Printer	

EXPANDED MEMORY BOARDS (FOR AT'S)

WORK	DO NOT WORK
AST Rampage AT	
INTEL Above Board	

3. SYSTEM ASSEMBLY

3.1 Hardware Setup

The purpose of this section is to describe how to assemble your system. It assumes that your computer is already configured with at least the minimum requirements specified in Section 2.

1. Install National Instruments GPIB in an empty slot in the computer, if not already there. See National Instruments' manual for detailed instructions.

2. If you have a printer, connect it to the main parallel interface on the personal computer.

3. If you have any other peripherals (such as a hard disk) connect them per the manufacturer's directions.

4. It is recommended that you execute the diagnostics provided with the computer and National Instruments interface before proceeding to ensure that they are correctly assembled and working.

5. Set the GPIB address of the LeCroy 8901A (or 6010) controller to the desired GPIB address (any address between 1 and 15).

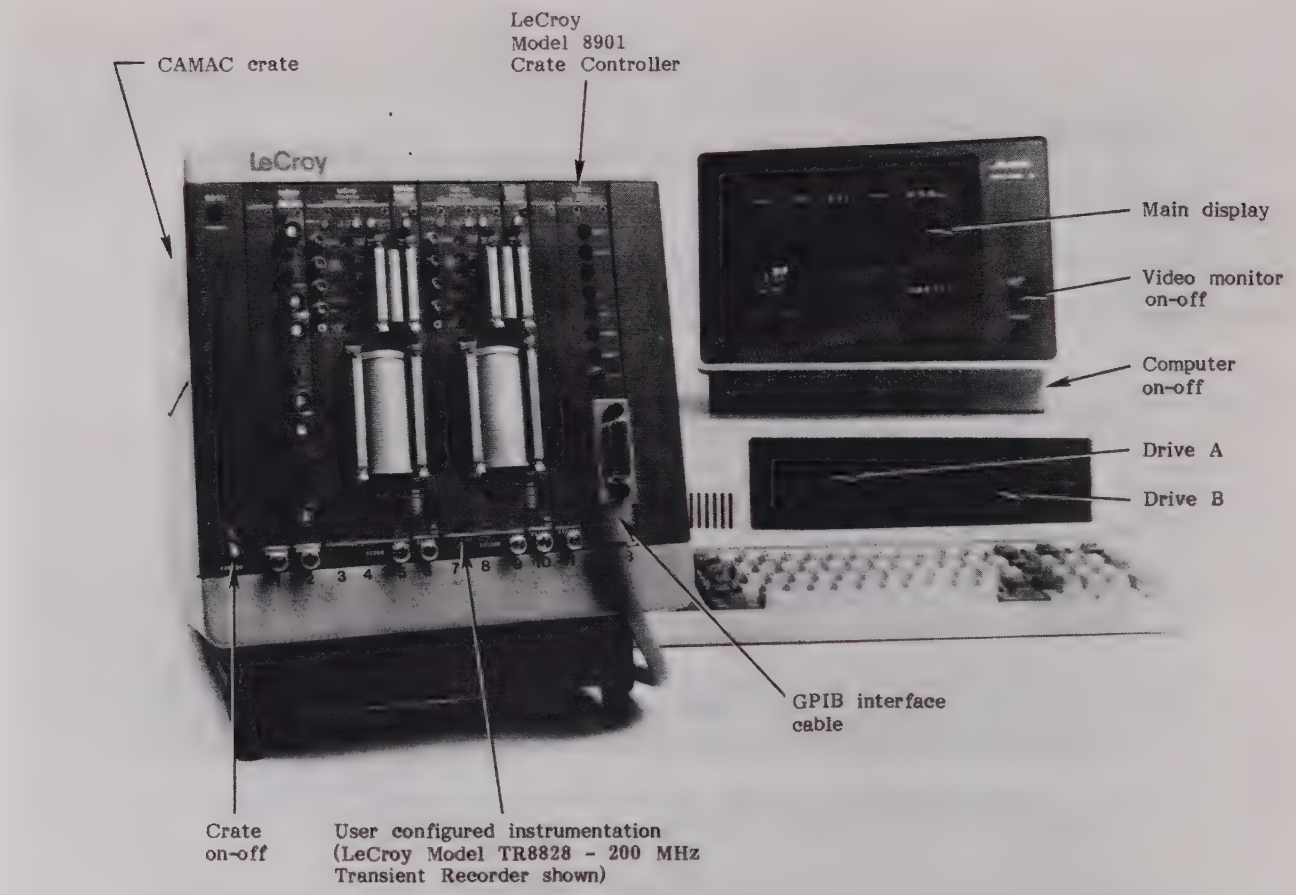
6. Install the Controller in the highest numbered slots in the CAMAC crate (Instrument Housing).

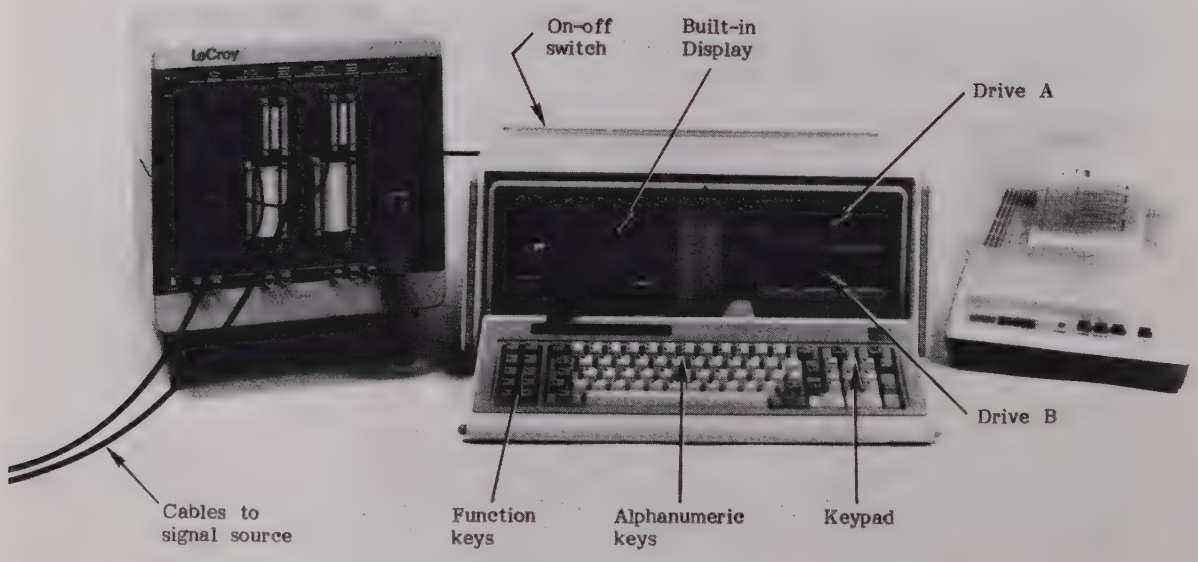
7. Install the rest of your CAMAC instruments (e.g., LeCroy TR8828C digitizer) in the crate. These can be placed in any slot.

8. Connect the GPIB cable from the connector on the front of the LeCroy Controller to the connector on the National Instrument's GPIB board. (Rear of the PC)

9. Connect the signal and trigger (if desired) to your instruments.

SYSTEM 6900PC - IBM Personal Computer





SYSTEM 6900PPC - IBM Portable Personal Computer

Figure 3.1 System Identification

3.2 Setting up CATALYST Disks

These instructions detail how to set up your disks (either floppy or hard) so that CATALYST will automatically come up and run when you turn on your PC.

To setup CATALYST to run from floppy disks:

1. Power-on your computer with a DOS disk in drive A: (top)

2. Enter date and time when prompted to do so.

3. When DOS prompt **A>** appears, you must make a copy of the CATALYST Boot and Program disks. To do this:

   a. type **DISKCOPY A: B:**
   b. remove DOS disk and insert CATALYST Boot Disk in Drive A
   c. insert blank disk in Drive B and press any key
   d. When copy is complete, answer **Y** to the question "copy another?"
   e. remove both disks and label new disk CATALYST Boot Disk
   f. insert CATALYST Program Disk in Drive A and another blank disk in Drive B and then press any key
   g. When copy is complete, answer **N** to more copies?, remove both disks and label new disk CATALYST Program Disk.

4. Put new copy of CATALYST Boot Disk in Drive A.

5. Type **SETUPCAT B:,g**
   where g = 2 for National Instruments GPIB 2 card
   2A for NI's GPIB 2A or IBM's GPIB card
   3 for NI's GPIB 3 card.

6. Following directions printed on the screen.

7. Once setup is complete, you can 'boot' the computer (power on or press the CNTL, ALT, and DEL key simultaneously) with the CATALYST Boot Disk in Drive A, following the directions on the screen.

To put CATALYST on your hard disk:

1. If the disk is not formatted or the operating system is not on the disk, see your DOS manual.

2. Power on the computer.

3. When DOS prompt **C>** appears, put the CATALYST Boot Disk in Drive A, and type **A:**. The DOS prompt should now be **A>**.

4. Type **SETUPCAT C:,g**
   where g = 2 for National Instrument's GPIB 2 card
   2A for NI's GPIB 2A or IBM's GPIB card
   3 for NI's GPIB 3 card.

5. Follow directions printed on the screen.

6. Once setup is complete, you can remove this disk and 'boot' the computer (power on or press the CNTL, ALT, and DEL keys at the same time) from the hard disk and CATALYST will automatically run.

4. GETTING STARTED

The purpose of this section is to give you an overview of the **CATALYST** features. A detailed description of all its commands can be found in Section 5.

To start **CATALYST** when the computer is powered off:

STEP 1: Turn on the CAMAC crate and any other peripherals (printer, mass storage device, etc.).

STEP 2: Turn on the PC.

STEP 3: For a PC with floppy disks only: insert the disk labeled CATALYST BOOT Disk in Drive A (top). (If you have not run SETUPCAT, see Section 3.2 now). When the power on test is completed, you will be instructed to insert the CATALYST Program Disk and press any key when ready. CATALYST will then be loaded into the PC. When CATALYST's Main Display (figure 4.1) appears on the screen, you may begin.

For hard disk systems:
If you have used SETUPCAT (see Section 3.2) to install CATALYST on your hard disk, CATALYST will be loaded automatically after the PC completes its power on tests. Once loaded, CATALYST prints the main display (Figure 4.1) on the screen and awaits your command.

If your computer is already powered-on, and DOS is loaded, you can start **CATALYST** as follows:

STEP 1: If the PC has a fixed disk, type **C:**, press ENTER (<—┘) and then type CD**CAT** plus ENTER. If system has only floppy disks, place the **CATALYST** Program Disk into Drive A.

STEP 2: Power-on the rest of the equipment.

STEP 3: Type **CATALYST.** Here again, when the Main Display is seen, the system is ready for your commands.

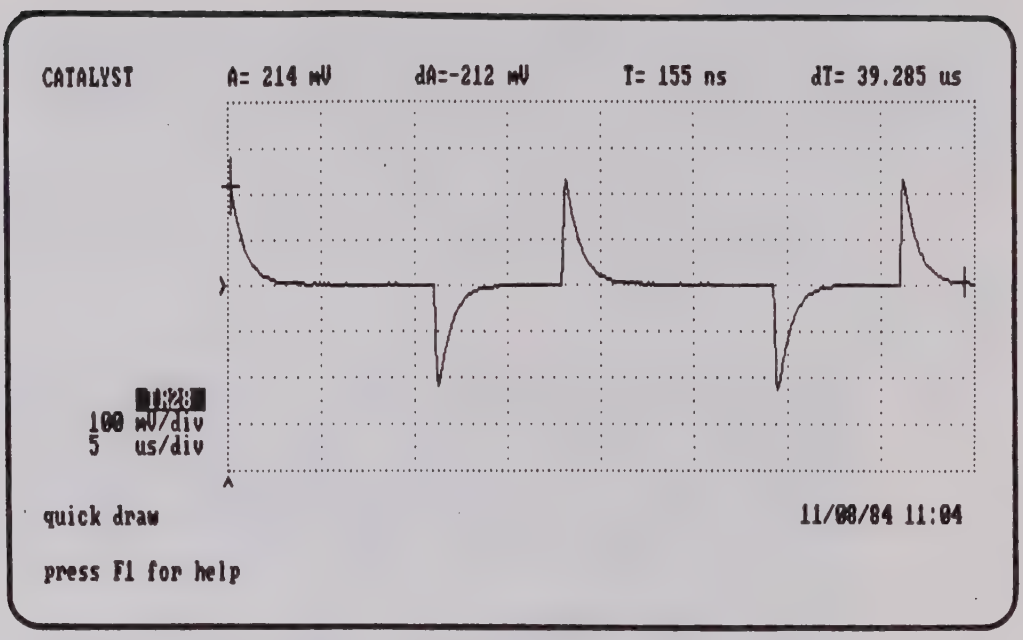
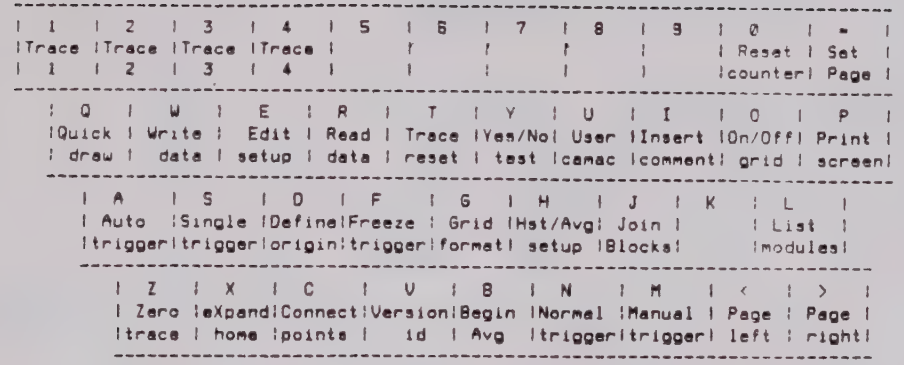


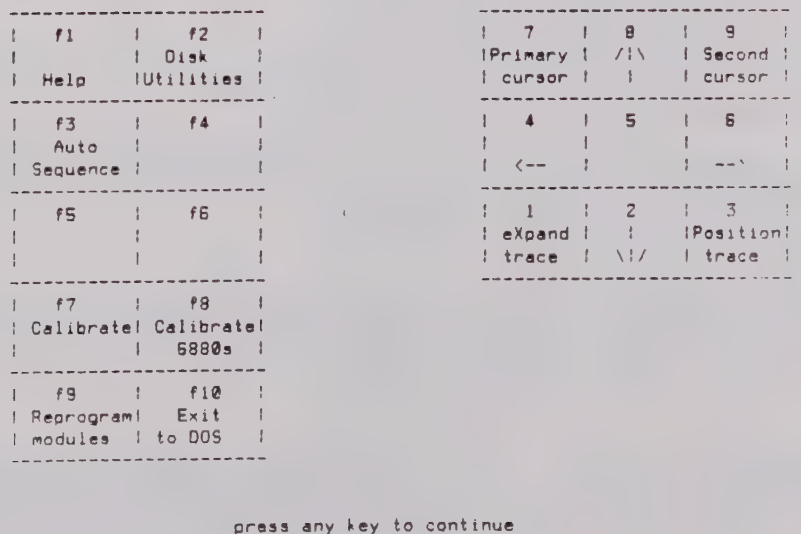
Figure 4.1 CATALYST'S main display

If you don't know what commands are available, you can press the "Help" function key labeled 'F1'. This draws Figure 4.2 on the screen to remind you of all the primary commands available. Pressing any key other than the space bar causes Figure 4.3 to be displayed. This shows what the function keys and the keypad are used for. To return to the Main Display from either of these "Help" screens just press the space bar. To enter any of the commands listed on the menu, simply press the key corresponding to it after returning to the main display. If more information is needed, the system will prompt you for it. Otherwise, it will immediately perform the function. (Note that some commands may take several seconds to complete. Since the commands are buffered, you can enter several commands while waiting.)



press space bar to return or any other key for more information

Figure 4.2 Help screen showing primary key functions



press any key to continue

Figure 4.3 Help screen showing function keys and keypad

Most of the primary commands in the system are single keystrokes; that is, only the key corresponding to the first letter of the command must be pressed in order to execute that function. For instance, if the modules in your system are configured properly, pressing **A** selects auto trigger mode which repeatedly arms and triggers the modules. For some commands, however, additional information is required, in which case a message is displayed stating what information is needed. You must then type the needed information and press ENTER. An example of this type of command is Write data. When **W** is pressed, the message "enter file name:" is displayed. For still other commands, since there are several possibilities from which to choose, a new list (or menu) of commands is displayed at the bottom of the screen. While a menu is displayed, any command in the list may be entered by pressing the key corresponding to its first letter. Pressing the space bar erases the command menu and returns the keys to their primary functions. **G** is an example of this command type since it activates the grid format menu.

The following example illustrates a sequence of CATALYST commands used to acquire and display data from a LeCroy TR8828C 200 megasample/sec digitizer which has been placed in slot 6 of the instrument mainframe. The GPIB device address on the mainframe is 12.

Install WAVEFORM CATALYST to your floppy or hard disk using Section 3.2 of the manual

Turn instrument mainframe and computer on.

Load WAVEFORM-CATALYST as describe above.

When CATALYST is loaded choose digitizing parameters as follows:

COMMAND	RESULTS
List modules	The list of modules in the system is printed along the bottom of the screen. Select number of module you want to operate
3. 8828	Setup menu will appear. Use arrow keypad (left, right, up and down) to move reverse video to digitizing parameter your wish to adjust. Press Enter on menu and select desired parameters. E.G.: NAME CH1 ACQUIRE Yes GPIB ADR 12 SLOT 6

Choose sampling rate, memory length, pretrigger, etc. Press Help for explanation of TR8828C parameters.

You may store this setup using Write in the menu. Selecting the file name sd8828.cat causes the current menu values to be loaded automatically each time CATALYST is begun. Pressing the space bar exits the setup menu.

To select which trace to display:

1 Trace Number	Traces are numbered from 1 to 4 from the bottom to the top of the screen. CATALYST prompts for name of channel to be displayed.
CH1	Enter four digits corresponding to name of module as defined in setup menu.
Grid;	Select how many grid boxes you want (Single, Dual or Quad). If you have multiple traces displayed, this function permits you to display them separately or overlayed.

Connect input signal and trigger input (STOP on TR8828C)

Select CATALYST triggering mode

Auto	Data will be displayed from TR8828C with parameters selected. Trace may be expanded, positioned and operated upon with cursors. Reverse video on channel name (at left of screen) indicates which trace CATALYST keys are operating on.

Note that if you have more than one channel operating, each memory must be read out before any recorder can be rearmed. The active trace is selected by moving the primary cursor (selected by pressing the 'Home' key) up and down.

To turn off an undesired trace, press the trace number you wish to remove. When prompted for trace name, press return without entering any ID.

The screen should now be updating with data from the TR8828C. Other CATALYST functions are highlighted in the user's manual and CATALYST user's card.

Additional Help screens like the one pictured in Figure 4.5 are available for each module type in the List Module menu. To access these help files, enter the Setup Menu and press **HELP**.

Figure 4.4 is provided to give you a quick description of the various parts of the display.

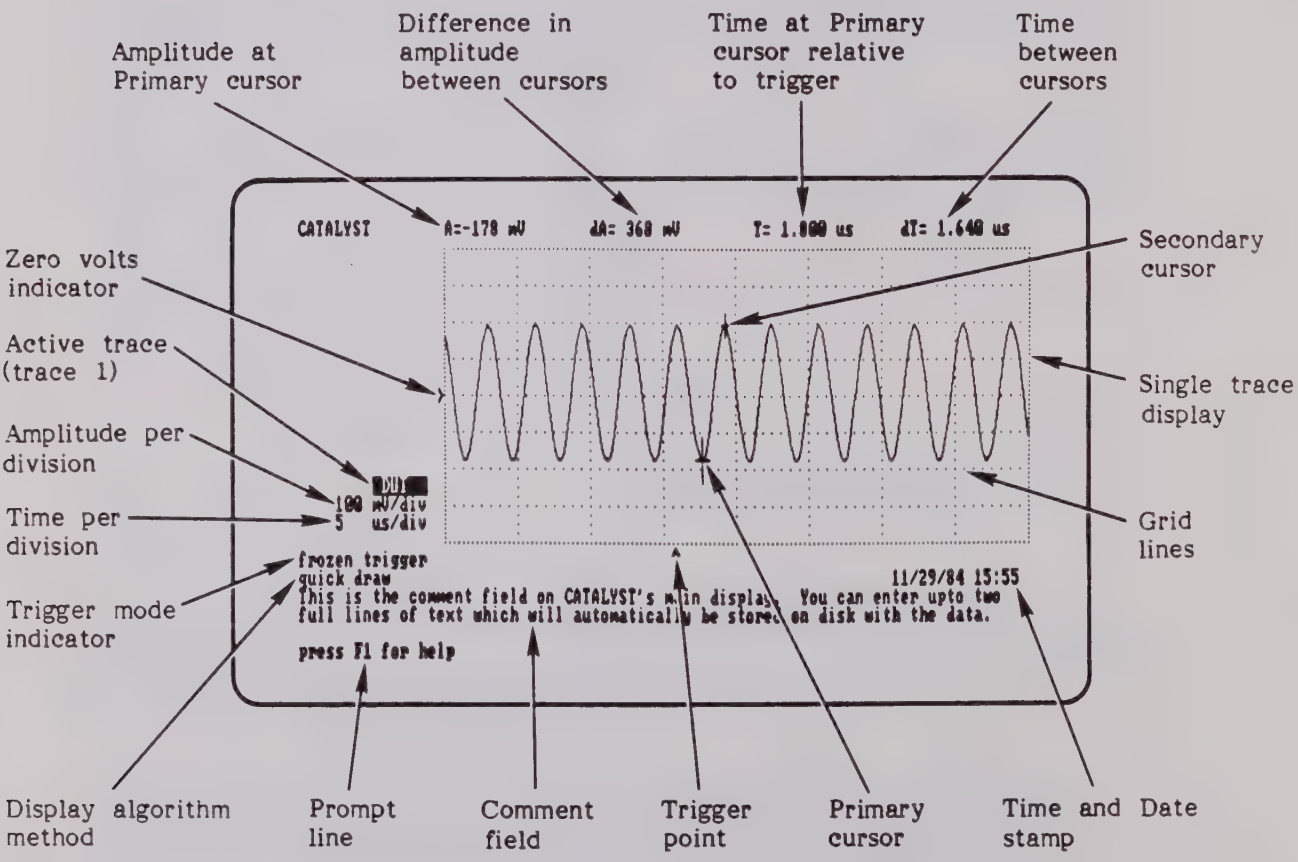


Figure 4.4 Parts of the main display

TR8828C - 200 MegaSample Transient Recorder

8-bit resolution
Upto 2M memory
512 mV peak-to-peak signal range

NOTES:

1. If an external clock is desired, the PERIOD must be set to EXT and a 200 MHz clock must be used.

2. Offset range is 256 mV (unipolar positive signal) to 256 mV (unipolar negative signal).

3. Full scale volts is provided as a correction factor for amplitude in case an amplifier or attenuator is used. Valid range is .1 mV to 100 V.

4. The first and last 16 samples may have been overwritten and are therefore set to zero so that erroneous data will not be displayed.

5. Pretrigger sample size (PTS) ranges from 0 to 8 eighth's of memory.

6. Data is displayed in blocks of 8k words. Multiple blocks may be viewed simultaneously by assigning the module to more than one trace and paging to the desired blocks.

7. Memory size is selectable in 16K steps upto 2 megawords. It is possible to read in and store less than 16K points but 16K points will be sample.

Figure 4.5
CATALYST **HELP** Screen for Model TR8828C 200 Megasample/sec Recorder

5. COMMAND DESCRIPTIONS

This section describes what each of the commands do. They are grouped by the function they perform.

The first group of commands is there to aid you in setting up the instruments to meet your needs. The next subsection discusses the various ways to start and trigger the instruments. The third group of commands is for specifying how the data is to be displayed. Following that is a section describing the uses of the keypad. The fifth section discusses how to get a hard copy of the screen, while the sixth talks about the disk utilities. The last section contains information on several additional commands.

5.1 Instrument Commands

5.1.1 Setup Menu

There are three ways of setting up or programming the instruments in your system. The first way is by pressing List module types on the main menu. The module type directory (Figure 5.1) is displayed, showing all of the module types for which the program is configured. From this list, select the number corresponding to the instrument you wish to program. The parameter table for that instrument is printed on the screen along with the setup menu. (Figure 5.2). On the table, there is an inverse video cursor indicating which parameter is being edited. Using the arrow keys on the keypad you can move the cursor right, left, up, or down to get to the parameter(s) you wish to change. To modify the current parameter, use the commands listed in the menu. When you have made all of your changes, pressing the space bar causes all of modules whose address (GPIB address and slot) is specified to be reprogrammed with the new values and returns you to the main menu. Reprogramming of the modules in the system can be accomplished by pressing the F9 function key.

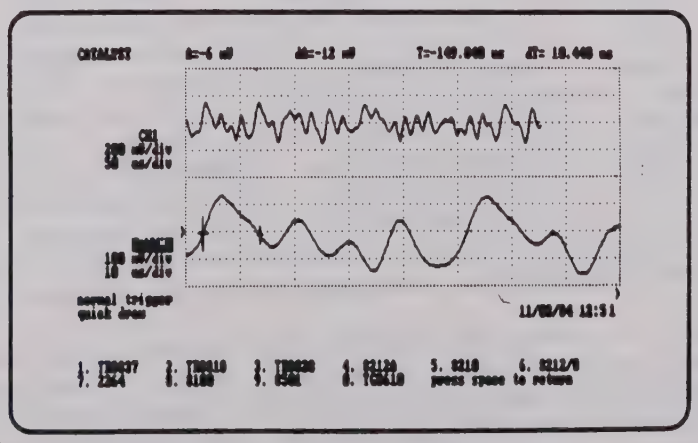


Figure 5.1 Module Type Directory

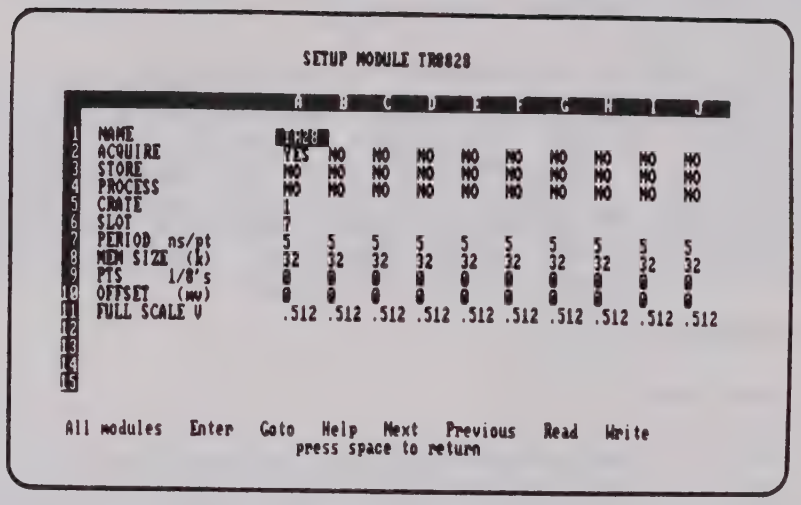

SETUP MODULE TR8828
A B C D E F G H I J
1 NAME TR28
2 ACQUIRE YES NO NO NO NO NO NO NO NO NO
3 STORE NO NO NO NO NO NO NO NO NO NO
4 PROCESS NO NO NO NO NO NO NO NO NO NO
5 CRATE 1
6 SLOT 7
7 PERIOD ns/pt 5 5 5 5 5 5 5 5 5 5
8 MEM SIZE (k) 32 32 32 32 32 32 32 32 32 32
9 PTS 1/8's 0 0 0 0 0 0 0 0 0 0
10 OFFSET (mv) 0 0 0 0 0 0 0 0 0 0
11 FULL SCALE V .512 .512 .512 .512 .512 .512 .512 .512 .512 .512
12
13
14
15
All modules Enter Goto Help Next Previous Read Write
press space to return

Figure 5.2 Setup Menu and Parameter Table

The first six rows on the parameter table are the same for every module in the system. They define parameters which are used throughout the **CATALYST** program. The first parameter, the Module Name, must be a unique four-character identifier which is used to refer to the module when assigning it to the display, acquiring data from it, or storing data to disk. The second parameter, ACQUIRE, must be enabled if data is to be acquired from the module, whether or not the module is to be displayed. STORE, which is the third parameter, allows you to specify the data to be automatically stored to disk each time the module is triggered. The next parameter, PROCESS, is used to enable processing of the data. Currently a 'sine fit' routine is supplied (see Appendix D) but you may make your own (see **CATALYST** Function Library manual for details). The next two parameters specify the module address. GPIB ADDR is the GPIB address (1-15) of the LeCroy controller in charge of the module, while the Slot is the position of the module within the housing. The rest of the rows in the matrix contain information which are used to set up the instruments and, hence, are specific to each module type.

If AUTOSTORE is selected CATALYST will automatically store data in all enabled 'write formats' (see Section 5.8). The names for AUTO STORE files are automatically generated. Regardless of format, the first character of the filename is the first character of the Module Name; therefore, the module name MUST begin with an alphabetic character. CATALYST format filenames are formed as: <module name>.<event count>. Example: TR37.001
ASYST™ format filenames are formed as:

"<module name>A<event count> [.<channel number>]"
Examples: TR37A001 (single channel)
T210A001.1 (first channel of multi channel acquisition)

ILS filenames are restricted to two alpha characters and four digits with no extension. Therefore, ILS format filenames are formed as:

<first character of module name><last character of module name>
<event count> <0 or channel #>
EXAMPLE: CHNA, event count = 1 becomes CA10.

Note that if you manually store a CATALYST data file with a filename of the form <four characters or less>.<extension> then the filenames for the other formats, if enabled, will be automatically generated as above.

The commands of the setup menu are described below:

All Modules - Sets the entire row to the value specified by the cursor. This is a quick way of setting all the modules of this type to the same value for a given parameter.

Enter Value - Prompts you to type in a value for the current parameter. The value cannot be more than four characters and must be followed by ENTER. If the parameter has defined settings, the value you entered must be among them or else the parameter is not changed and an error is reported.

Goto Parameter - An alternative to using the cursor keys. When pressed, the prompt "Enter row, column" is displayed. You must type the row number followed by a comma and then the letter of the desired comma. Hitting ENTER causes to cursor to move to the specified parameter.

Help - Erases the screen and displays information about this module type. Pressing any key returns you to the setup menu.

Next Value - Causes the next value in its list to be displayed if the current parameter has defined settings. For example, if the cursor is currently in the row for slot number, and the current value is 3, entering this command will change it to 4. If the parameter does not have defined settings, as with full scale volts (for an TR8828C) then you are prompted to enter an increment to the current value. The increment may be a positive or negative number.

Previous Value - Same as Next, except it changes the current value to the prior one from the list of defined settings. Using the above example, if the current value for slot number is 3 and 'P' is typed, the value is changed to 2.

Read Set-up - Allows you to read from disk a setup configuration which had been previously stored there. When the command is executed, you are asked to supply the name of the file to be read in (you may specify the drive as well e.g. A:MYSETUP.A28). The file is then read in, updating all the parameters for each module on this matrix. As always, when you exit this menu, the modules are reprogrammed with the new values.

Write Set-up - Allows you to save the entire matrix on disk just as you see it, so that it may be recalled later, using the read command. When you enter the command, you are asked to supply a file name. This can be any valid DOS filename ([drive:]name[.ext]) excluding path names.

0 thru 9, -, +, & . - Typing any of these characters when this menu is displayed is a shortcut way of typing in a parameter value (i.e. you need not press Enter first). Once the character is typed, you are prompted to supply the rest of the value followed by ENTER.

5.1.2. Edit Menu

The second way to change a module parameter is by using the Edit setup command. This command, which brings you to the edit parameter menu, prints the parameters for the active trace underneath the grid (Figure 5.3). This allows you to change the parameters while viewing the traces. Therefore, you can use this only for the modules which are displayed. The menu is actually a subset of the setup menu and includes the Enter, Next, and Previous, commands. Like the setup menu, typing a number, plus, minus, or decimal point is equivalent to first pressing Enter. The left and right arrow keys allow you to move the cursor between parameters. The up and down arrows change which trace is active. As a result, you can edit the new trace's parameters.

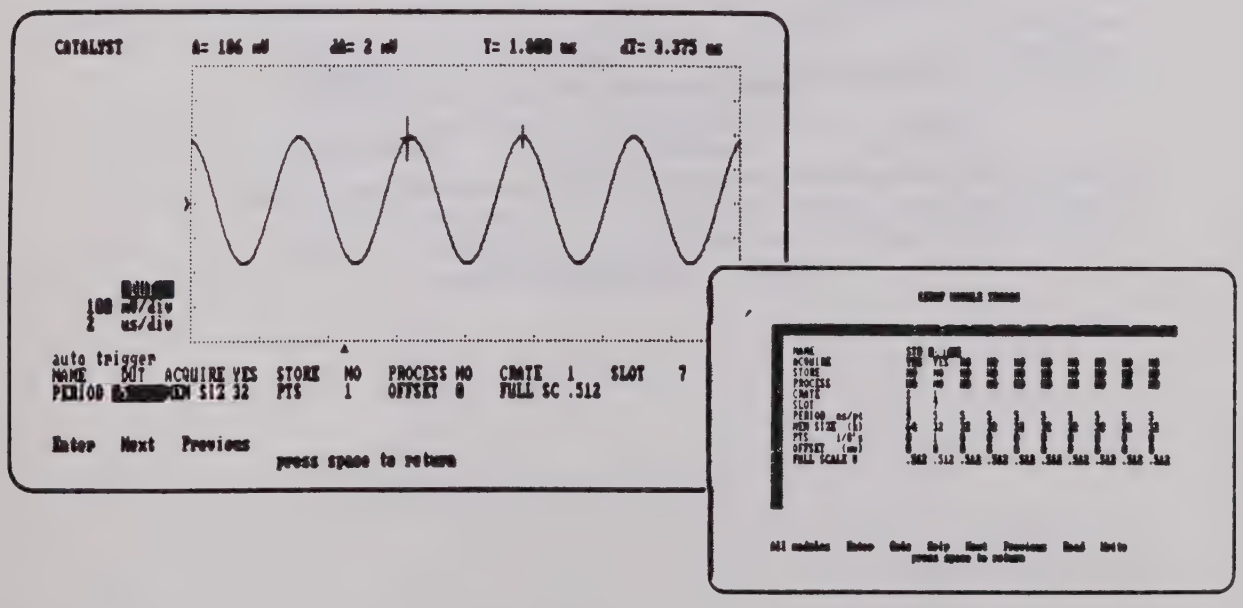


Figure 5.3 Edit Parameter Menu

5.1.3. User CAMAC Menu

The third way to control your instruments is with the User CAMAC command. This feature, which is useful for controlling unsupported instruments or performing a quick test of an instrument, provides nine registers in which you can enter C,F,N,A, (plus data) CAMAC commands. In addition, the registers may be stored or recalled from disk to permit execution of pre-configured commands.

When you enter the User CAMAC function (by pressing U) **CATALYST** prompts you with the message "User command (#,R,W,L)?". If you type R or W you will be asked to supply a file name from which the registers will be recalled or written to respectively. Typing L causes the contents of the registers to be displayed (until you press the space bar). Entering a number (1-9) displays the User CAMAC menu along with value of register you select as seen in Figure 5.4.

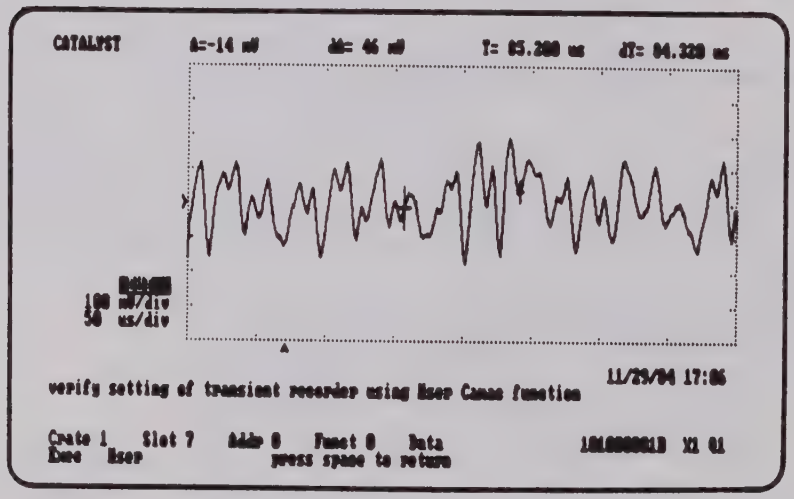


Figure 5.4 User CAMAC Menu

The commands of the CAMAC Menu are explained below. As with other menus, pressing the space bar returns you to the primary command level.

Crate - Permits you to enter the number of the crate in which the module is housed. (The crate number is the GPIB address of the LeCroy controller).

Slot - Allows you to specify the instrument's position (N) in the crate.

Address - Lets you give the module's sub-address.

Function - Allows you to enter the number corresponding to the function that you want to send to the instrument.

Data - Allows you to specify what data should be sent to the module when a CAMAC write command (F > 15) is executed. Up to 24 bits of data are entered in either binary or decimal by typing the number followed by a **B** or **D**, respectively. If running an autosequence, you can also use any one of the autosequence variables by typing its name (V1,...,V25) in place of an actual number. When **Execute** is pressed, the current Value of the variable is substituted. (See Section 5.6 for more details.) When a read command is executed, the data read from the instrument is displayed in the data field in either binary or decimal, based on the current setting (B or D) of the field.

Execute - Each time **E** is pressed, CATALYST will send the F, A and data (if F > 15) to the module specified by C and S and display status (X and Q) response of the instrument next to the data. If it is a read command (F < 16) 24 bits of data will be read and displayed.

User - User command works just as it did on the primary menu. You can enter a new register number, read or write to disk, or display a list of the current register values.

5.2 Trigger Commands

The commands involved in data acquisition are similar to those on an oscilloscope which allow you to select the trigger mode. These commands apply only to modules which have been armed for acquisition on the setup menu.

Auto Trigger - In this mode, the modules which had been selected for acquisition are armed and a one-second timer is started. When a trigger occurs, the data is read and stored or displayed as specified and the module is rearmed. If however, the timer runs out before the modules are triggered externally, the system stops the modules, reads the data, and approximately two seconds later restarts them.

Freeze Trigger - This is a toggle function. If a trigger mode is active, when this is pressed, the mode is disabled. Pressing it again restores the prior trigger state.

Manual Trigger - Allows you to trigger all the armed modules from the keyboard.

Normal Trigger - Arms the selected modules for acquisition and waits for them to be triggered. As each module is triggered, the data is read into the computer for display and/or storage. After all the modules have been triggered, the sequence is repeated.

Single Trigger - Arms the selected modules for acquisition and reads the data when they are triggered. It does not rearm them.

Begin avg/hist - Starts averaging and/or histogramming the selected trace. (See Section 5.5 for more details).

5.3 Display Controls

You may choose to display the data from up to four modules at one time. The commands described below control the manner in which data is presented on the screen. When data is displayed, it is converted to engineering units and scaled both horizontally and vertically. The amplitude per division for each trace is displayed directly below the module name for the trace as seen in Figure 5.5. The time per division is displayed on the next line.

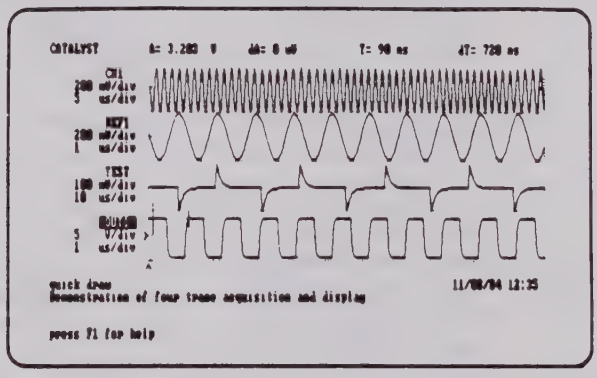


Figure 5.5 Quad Grid Display Without Grid Lines

1, 2, 3, & 4 - To assign a module to a trace, type the number corresponding to the desired trace (1 is the bottom trace, 4 is the top one) see Figure 5.6. You are then asked to enter a label for the trace. If you just press ENTER, the current trace is deleted. If you type in a label and press ENTER, all the parameter tables are searched. If your entry is found, that module is assigned to the trace and whenever data is acquired from the module it is displayed according to the trace settings. You can assign a module to more than one trace, which allows you to display the data in different ways (e.g. one unexpanded and one horizontally expanded) or if there are multiple pages or channels of data you can display up to four pages simultaneously. Each copy of the data will be updated when a trigger occurs. Alternately, if you wish to temporarily save data (without writing it to the disk), you can do so by entering a name for the trace which is not used as a module name. In this case, **CATALYST** will then ask you to supply the number corresponding to the trace you want saved. The current data will be copied to the new trace. When a trigger occurs, the original trace is updated but the saved one is not. This allows you to compare a trace with subsequent ones.

Connect points - When a trace is drawn, you have the option of having the points connected by line segments or just displaying the actual points. This command toggles between the two options. (Note that when a trace is horizontally expanded such that more than two interpolations are necessary between points, only the actual points will be displayed.)

Define origin - Allows you to redefine the beginning point of the active trace to the location of the primary cursor. If the trace is not horizontally expanded and there is more data available, a new data block is read in. Pressing the page left key, resets the origin, while the page right key causes the next block (if available) to be read.

Grid format - Brings up the grid format menu which has three entries: Single, Dual, Quad. Single selects one grid on which all of the traces are plotted, one on top of another. Dual selects a format with two half-size grids. In this format, traces 1 and 2 are plotted in the lower box, while 3 and 4 are drawn in the upper one (see Figure 5.6). Selecting Quad allows each trace to be displayed in a separate quarter high grid as in Figure 5.5. Pressing any of the choices takes you out of the menu after changing the format (press the space bar to return without changing the format).

Histogram/average Brings up the histogram/average setup menu. See Section 5.5

Insert text - Allows you to type two lines (160 characters) of annotation, which are displayed on the screen. This information is also stored on disk each time data is written. When a comment is typed, pressing ENTER saves the text and allows you to enter additional commands.

Join Blocks - When the primary menu is active, and J is pressed, all of the data from the module corresponding to the active trace is read, processed by the anti-aliasing algorithm, and then displayed. Moving the cursor to the point of interest and redefining the origin causes the block of data containing the new origin to be displayed. Note this feature should not be used with instruments that have less than 1K words of data.

On/Off grid - Toggles the grid lines on and off. Figures 5.5 and 5.7 compare the display with and without the grid lines.

Quick draw - When data is read in from a module, there can be as

many as 8192 points. Since the width of the grid is 512 points, it is impossible to display each data point. In order to represent the data as accurately as possible, an anti-aliasing algorithm is used. However, on large blocks of data, this algorithm is slow and an option is provided which displays the data without the algorithm. To switch between the two drawing methods you need only press **Q**. NOTE: with either method, any data point exceeding the instruments' full scale is drawn at the edge of the grid.

eXpand on-off - If you have expanded a trace horizontally and wish to unexpand it so as to see the whole trace, just press **X**. This saves the current settings and then resets the trace. That way, if you press **X** again, the trace is redrawn in the expanded format.

Zero position - If you have used the position keys to move the trace around, pressing **Z** will restore the trace back to its initial position.

< & > - When the trace is not expanded, these keys allow you to look at the next block of data (if any) in the direction of the arrow. Each block of data can be up to 8K bytes long. For a module that has 32K bytes, therefore, there would be four blocks. When a trace is expanded, you can see only part of the block of data. These keys now let you scan the various "pages" (subsections) in the block, each time showing the same size page.

= - Set block number. If the trace is not expanded, the range of available blocks is shown and the user is prompted for a choice. If a valid response is given, that block is displayed. Invalid responses cause no action. No Response is generated if only one block of data is available.

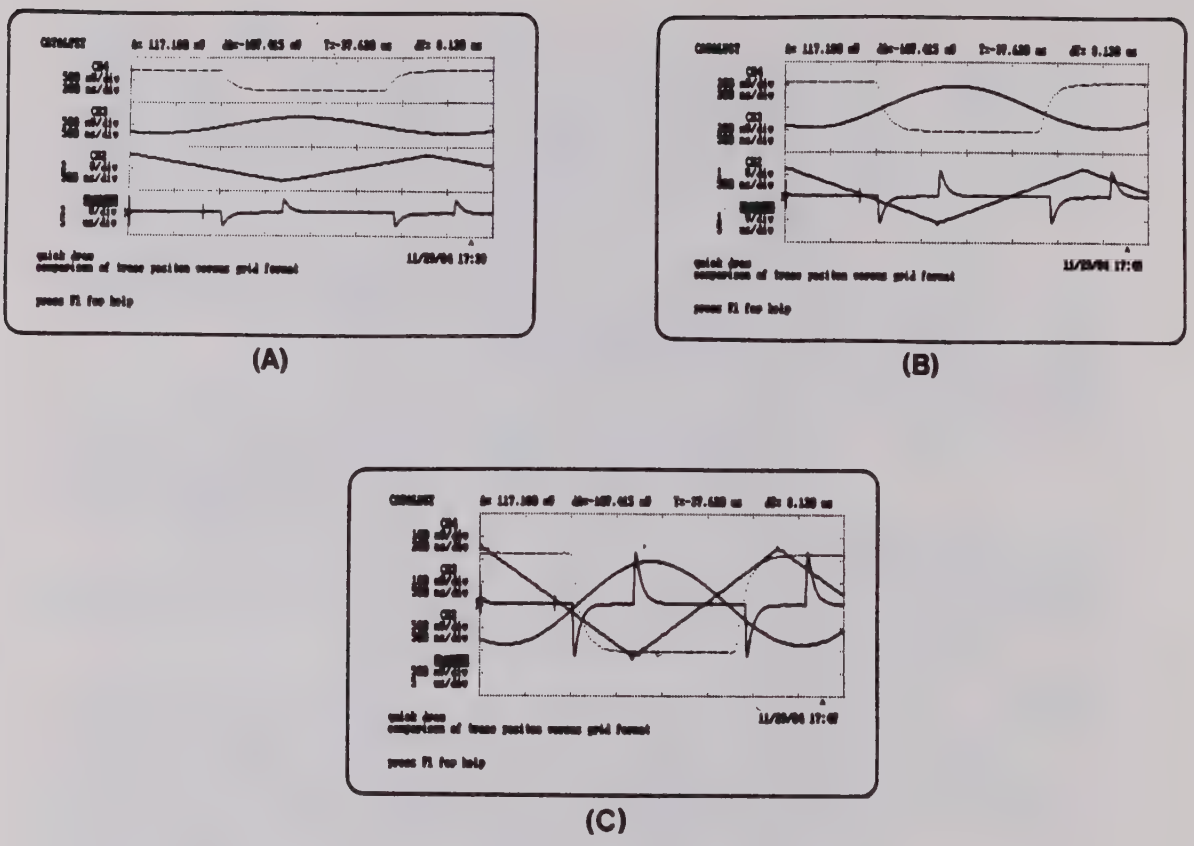


Figure 5.6 Position Of Trace As A Function Of Grid Format

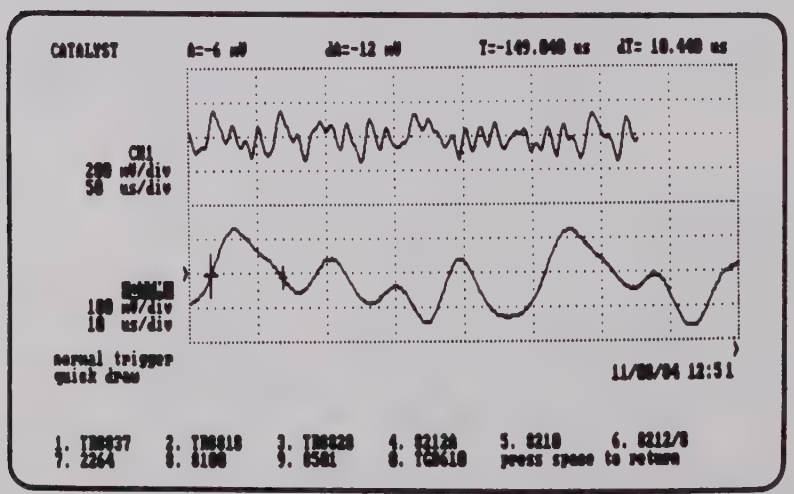


Figure 5.7 Dual Grid Format with Grid Lines On

5.4 KEYPAD FUNCTIONS

The arrows on the keypad are used for several different functions, since they are a convenient way of indicating direction. Their function is determined by the one last selected. For example, when the system is started, the arrow keys are set to the primary cursor function. Thus, pressing the right arrow will move that cursor right. If you then select the secondary cursor, pressing the left arrow moves the secondary cursor left and the right arrow moves it right. Each of the four functions affect only the active trace (the one whose name is highlighted in inverse video). They are described in detail below.

5.4.1 Trace Cursors

There are two cursors on the active trace. The larger one (on the left edge initially) is referred to as the primary cursor. It's position is used to compute the values of amplitude and time which are displayed at the top of the screen. When the Primary cursor function is selected, the left and right arrows move both cursors along the trace in the corresponding direction and the measurements are updated. The up and down arrows move the cursors to the trace above or below, respectively, thereby changing which trace is active. The other cursor is selected by pressing the 'secondary cursor' key on the keypad. The time and amplitude between the primary and secondary cursors is also calculated and displayed at the top of the screen. When this function is selected, the right and left arrows move the second cursor along the trace while the up and down arrows move between traces as before. Note that if the primary cursor is active, the secondary cursor moves along with it (except when it reaches the screen boundaries). For both cursors, each time the left or right arrow is pressed, the cursor is moved one step in the corresponding direction. If either key is held down, however, the cursors move in increasingly larger steps until the button is released or the cursor reaches a boundary.

5.4.2 Expand Trace

The Expand Trace function allows you to look at a particular area of the active trace in more detail. When this mode is active, the up and down arrows control the vertical expansion, while the left and right arrows control expansion horizontally. Specifically pressing the up arrow expands the trace outward from the zero line by a factor of two. If it is vertically expanded, pressing the down arrow decreases the trace by a factor of two. When the right arrow is pressed, the trace is stretched by a factor of two from the origin. Pressing the left arrow performs the exact opposite results.

5.4.3 Position Trace

When the Position Trace key is pressed, the arrow keys may be used to move the active trace around on the screen in the corresponding direction. As with the cursors, each time the key is pressed the

trace is moved by one pixel in that direction, whereas holding down the key causes the trace to move in progressively larger steps. The Position Trace feature allows you to line up one trace on top of another for instant comparisons.

5.5 Averaging and Histogramming

CATALYST can average records from a Transient Recorder or it can form a histogram (curve showing distribution of amplitudes) of a point (or points) of a selected data record. Up to four traces may be averaged or histogrammed at one time. CATALYST has an 8K buffer (31 bits wide) available for averaging and histogramming. This buffer is divided equally among all of the traces enabled for histogramming and/or averaging (i.e., 8K for one trace, 4K for two traces, etc.). The averaged or histogrammed data may assume values of up to 2**31. That means for an 8-bit transient recorder, it is possible to average more than 8 million sweeps, or over 2 billion acquisitions may be histogrammed.

When the LeCroy 6010 controller is present, very fast summation and exponential averaging is performed by the 6010 for LeCroy's high speed transient recorders, Models TR8837F, TR8818, and TR8828C.

In order for the 6010 to average the data from an instrument it must be able to recognize the instrument's type. The 6010 does this by reading the ID code from each instrument when the instrument housing is powered on. Therefore, the ID code of TR8818's must be set to 1, TR8828C's must have an ID code of 2, and TR8837F's must have an ID code of 3. Refer to each instrument's manual for information on setting the ID code.

All of CATALYST's display manipulation (expand, position, etc.) controls may be used on histogrammed and averaged traces. In addition, the cursors provide the ability to make relevant measurements.

Time records to be summed must be synchronized to some signal, or the result of the average will be zero. Similarly, data to be histogrammed is usually acquired only at some event of interest. Therefore, CATALYST acquires data for averaging/histogramming as if Normal triggering were in effect. That is, it waits for each module to stop sampling before restarting the acquisition of more data. Hence, an external 'stop' or trigger should be used. Modules can be triggered manually by CATALYST, but this is not suitable for averaging.

5.5.1 Averaging and Histogramming Menu

Pressing the Hist/avg setup key on the main WAVEFORM-CATALYST menu causes the "histogram/average setup" menu to appear on the bottom of the screen (see Figure 5.8). All functions of this menu are used to set up averaging parameters. It is not possible to change this setup while averaging is in progress. The function of each item is discussed below.

1, 2, 3, or 4	Prompts choice of Averaged, Histogrammed or Normal for the trace corresponding to the number pressed. At power up, these settings are defaulted to Normal. If the trace is to be histogrammed, it also prompts entry of which point(s) or channel(s) are to be included in the histogram. If average is selected, the prompt "sum or exponential?" will be displayed if the instrument to be averaged is one which the 6010 can average and the 6010 is present. If Exponential averaging is chosen, you are then prompted for the power of two of the decay constant. For example, a response of "4" would mean that the effect of a sweep has decayed to 1/e of its original impact in 2^4, or 16, sweeps.
Number of sweeps	Prompts you for a number between 1 and 2 billion. This is the number of sweeps that will be averaged or histogrammed before CATALYST freezes averaging (histogramming) by going to frozen trigger. The default is 100. After averaging has paused, pressing F continues the averaging for the same number of sweeps and then freezes it again.
Display rate	Prompts you for the number of seconds between forced display updates. This setting has an effect only if none of the displayed traces are Normal. The presence of a normal trace causes the display to be updated each time data is acquired for that trace. If all the traces are averaged or histogrammed, then averaging proceeds without updating the display for this number of seconds. The number of seconds may be zero (display each time) to 300 (display every five minutes). The default is 5 seconds.
Begin average	Clears the averaging buffers, begins averaging (histogramming) with the selected parameters and then exits the Histogram/average menu. The Begin command is also available on the main menu to permit the user to restart the averaging once the traces have been setup.

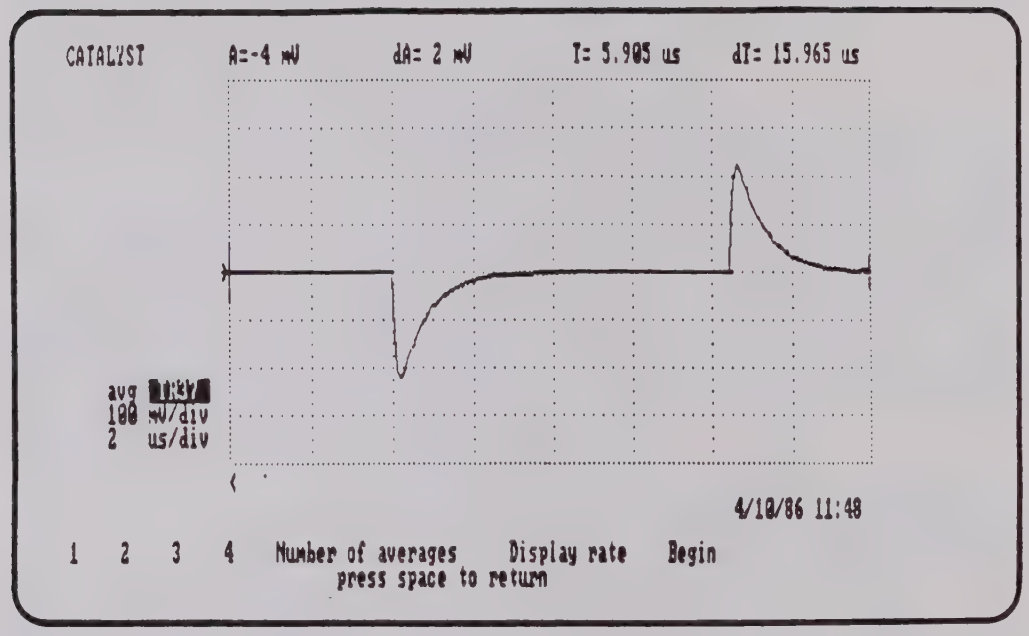


Figure 5.8: Histogramming/averaging setup menu

5.5.2 Example of Averaging

For this example, assume that trace 1 is labeled TR37 and that this label corresponds to a TR8837F that is enabled for acquisition in CATALYST's 8837 setup menu. The TR8837 has the signal of interest connected to its RECORD SIG input and the trigger signal connected to its RECORD STOP input.

Select Normal triggering by pressing **N**. If the trigger signal exists, data should be acquired and displayed. Verify that this data is what you want to average. Now press **H** and get to the histogram/average setup menu. Press **1**, then **A** and press ENTER to select trace 1 for averaging. The label 'avg' will appear to the left of the trace. Since this is the only trace selected, up to 8K points may be averaged. However, if less than 8K is acquired from the 8837, then only those points will be averaged. Press **B** to begin averaging and return to the main menu. A message indicating the number of sweeps taken appears under the trace display and will advance each time a new record is included in the average. Until the first display update, the display will remain blank. When the specified seconds have elapsed the display will be updated. You may press **Freeze** at any time; this will cause a display update and freeze averaging. Pressing **F** again will permit averaging to continue. When the number of sweeps selected have been averaged the display will update and trigger mode will go to 'frozen' automatically. You may press F to continue averaging after this point also. You may press B at any time to restart the averager.

To discard the averaged data and return trace 1 to normal, either enter the histogram/average setup menu, select the trace and choose Normal, or reassign the trace.

5.5.3. Example of Histogramming

Histogramming is performed similarly to averaging, as described above. After you select a trace for histogramming you will be asked which points you wish to include in the histogram. This is useful for modules which return one point per channel, in which case multiple channels can be included in the same histogram. It is also useful for transient recorders, so that many consecutive points from each acquisition can be included in the same histogram.

For a histogrammed display, the vertical axis is the number of occurrences of a particular measurement, and the horizontal axis is in whatever units the module measures, (Coulombs of charge for a charge to digital converter, seconds for a time to digital converter, or volts for a transient recorder or ADC, for example).

There is no 'quick draw' algorithm for histogrammed data. If there is more than one data point per pixel, the data values are summed together to show the number of hits in the wider display pixel. All other display controls function normally -- including trace positioning, origin redefinition and expansion.

5.6 Autosequence

CATALYST's autosequence allows you to teach the computer a series of commands which may then be saved on disk and executed again. All CATALYST commands discussed previously may be used as well as several "flow control commands" (For...Next, If, Wait) which are provided specifically for autosequencing. In addition, 25 variables are available for computation and control while an autosequence is executing.

The autosequence feature divides WAVEFORM-CATALYST operation into three modes: Interactive, Learn and Run. Interactive is CATALYST's normal mode. In this mode commands are accepted from the user via the IBM-PC's keyboard. Learn mode appears to the user to be identical to Interactive mode. Run mode accepts commands and responses from the stored autosequence instead of from the PC's keyboard. Exiting Run mode always returns to Interactive (normal) mode. Run mode may be automatically exited by the autosequence control commands WAIT and END, which are described below, or by pressing any key on the PC's keyboard. When the Main Display is drawn, the annotation "LEARN MODE" or "AUTOSEQUENCE" will appear beneath the traces if CATALYST is not in Interactive mode.

5.6.1 Autosequence Menu

All functions of the autosequence feature are accessed by pressing F3. If you are currently in interactive mode and select the third function key, the autosequence control menu is displayed (see Figure 5.9). The function of each of these commands is described below.

Learn - If current autosequence is valid and hasn't been saved on disk, you will be asked whether or not you want to overwrite it. A yes answer enables you to overwrite it. On entering Learn mode, the menu changes from the autosequence control menu to the autosequence Learn menu, described below.

Read - Loads an autosequence that was previously stored on disk and checks it for errors which would prevent execution. If there are none, the autosequence may be run. The user is prompted for the name of the disk file.

Write - Stores the autosequence currently in memory to a disk file. The user is prompted for the name of the disk file.

Go - Changes the mode from Interactive to Run if the autosequence in memory has no errors. The menu will immediately change from the autosequence control menu to the autosequence Learn menu. The context information is restored to what it was at Learn time. Execution begins with the first token in the autosequence. Execution may be

aborted by pressing any key on the keyboard.

Edit - This permits you to view, delete commands from, and insert commands into the autosequence currently in memory. When Edit is selected, the entire display will be replaced by the edit display. A menu with edit functions will replace the autosequence control menu until the editor is exited. When edit is exited, the autosequence is checked for errors which would prevent execution. If there are none, the autosequence may be run. A description of Edit functions can be found in Section 5.6.3.

Print var - Prompts you to enter the number of an autosequence variable (1 through 25) and prints its value on the screen. The value will remain on the screen until the next prompt so that you may print the screen showing this annotation. If no value is entered as the variable number, the prompt area is erased.

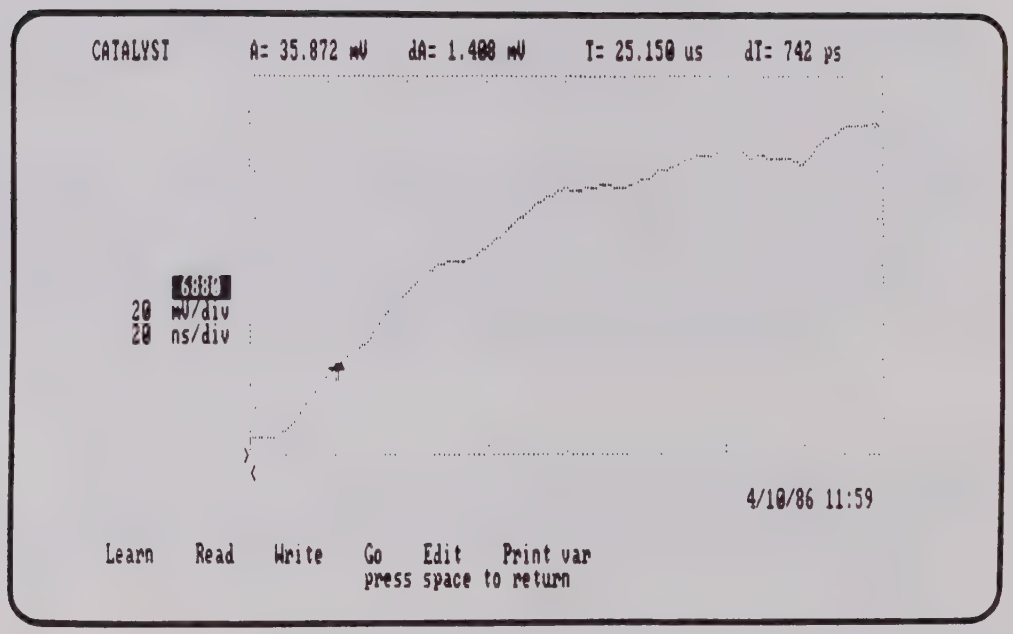


Figure 5.9: Autosequence Control Menu

5.6.2 Autosequence Learn Menu

Upon entering Learn mode, or on pressing F3 while in Learn mode, the autosequence Learn menu is printed at the bottom of the display (see Figure 5.10). Most of the commands on this menu are for controlling the flow of the autosequence. This allows you to assign variables, loop, branch on a specific condition, or pause the sequence until the occurrence of an event.

Assign - Prompts for the number of the target variable (1 to 25) and then for the left operand, operation if any, and second operand. The resulting assignment is of the form:

<target variable> = <operand 1> <operation> <operand 2> or, if no operation is entered:

<target variable> = <operand 1>

Permitted operations are +, -, * and /.
Permitted operands are A, dA, T, dt (the four cursor readout fields), UC (which is the last data read in with user CAMAC), variables V1 through V25, and LV1 through LV16 (the loop variables, see below), or any constant that the computer can represent as a real number. **(NOTE:** T and dT are always used to represent cursor horizontal position, regardless of the type of trace the cursor may be on.)

For - Begins a loop. Loops may be nested to 16 deep. The "depth" of this loop is displayed and the user is prompted from the desired number of iterations. The Loop Variables are available for inclusion in strings such as file names and user comments in two ways, as shown by the following examples:

F has no effect except when the autosequence is executed.

String in autosequence	
FILENAME.LV<number>	If <number> is less than or equal to the depth of the innermost loop but not zero, then the three characters LV<digit> are replaced by the value of the loop variable for the loop of nesting depth <number> and spaces replace any subsequent digits.

If the number is omitted or zero or greater than the depth of the innermost loop, the current loop variable is used.

%LV<number> — The number of characters used by %LV<number> (<number> may be omitted) is replaced by the value of the appropriate loop variable, determined as for .LV<number> above. Note that Edit must be used to embed %LV in the response to a filename prompt, since the % character is not valid as part of a DOS filename.

Next - Indicates the end of a loop. An error message is displayed if there is no open loop in the autosequence, in which case the **N** is not learned. When the sequence is executed, the loop is repeated the desired number of times. After the last time, execution proceeds with the command following the **N**. **Note:** A loop may end in a different menu than that in which it began. The "context" information is saved at runtime on execution of the **F**. Each execution of **N** restores this context, insuring that all commands have the same meaning for each iteration of the loop.

If - Prompts you for the left operand, comparison, right operand, and the label to jump to if the comparison is true at run time. The form of the IF statement is:

IF <operand> <comparison> <operand> then goto <label>
Permitted operands are the same as for Assign.
Permitted comparison operations are:

Any of these.....		Meaning
<	LT	Less than
>	GT	Greater than
<=	LE	Less than or equal to
>=	GE	Greater than or equal to
=	EQ	Equal to
!= <>	NE	Not equal

Label - Up to 10 characters, used as target of If. All

10 characters are significant. There are no restrictions on characters which may make up a label. Labels should be unique, as only the first of duplicate labels will ever be jumped to by an If. Learning extra labels (which no If accesses) is okay. See the warning in Section 5.6.5 about inserting labels when Editing an autosequence.

Print var - Prompts you to enter the number of an autosequence variable (1 through 25) and prints its value on the screen.

Wait - You are prompted for the type of event for which to wait. This can be new data, a specified number of seconds, the end of an average or histogram, or a response from the user. Wait has no effect until the autosequence is executed.

Note: Wait for data, Wait for average completion and Wait seconds drop out of run mode and into interactive mode. Any command may be performed. Entering Learn mode or Edit will cancel the wait. If the Wait is eventually satisfied, Run mode will automatically be re-entered and execution will proceed. When Run mode is automatically re-entered, the current menu is changed to autosequence Learn and the context information is restored, so execution continues in the same environment as when the W commenced.

'Wait for user response' does not drop out of run mode, but waits for the user to enter a number and/or press ENTER. Variable 25 is assigned the number entered by the user or zero if no number was entered.

End Learn - Puts an "End" marker into the autosequence and exits Learn mode. The autosequence learn menu is replaced by the autosequence control menu. The autosequence in memory is checked for errors which would prevent execution. If there are none, the autosequence may be executed by pressing Go.

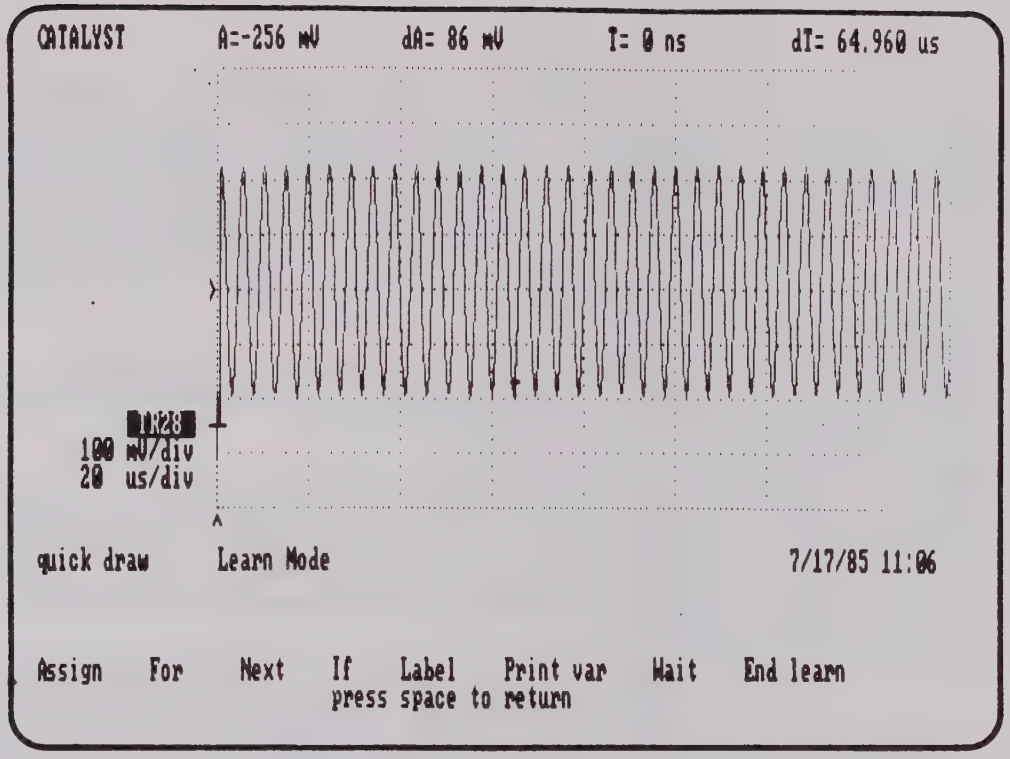


Figure 5.10: Autosequence Learn Menu

5.6.3 Autosequence Edit Commands

When the autosequence editor is entered, a display of 20 lines of the current sequence is shown. The eleventh line will be marked "PC>". This is the command pointed to by the autosequence "Program Counter". If you get an error such as "Expected Key" during execution, you may enter the editor and the PC will point to the offending token in the autosequence - probably a string where a single key command was expected.

The column of numbers to the left of the commands, but to the right of the "PC>" marker is the byte in memory at which this token starts. An autosequence may be up to 4000 bytes long (including the End marker).

On the display of commands, single key commands are shown by one or more characters not in quotes. Strings are shown with single quotes enclosing the string. Flow control constructs (For, Next, If, Wait, End, etc.) are shown enclosed in angle brackets.

The twenty-first line on the display shows the number of bytes used and the number of bytes available in the current autosequence. Figure 5.11 shows a typical display generated by Edit.

Edit commands effect the token at the PC or move the PC, as follows:

Up arrow (on keypad) - Backs up the PC one token, which corresponds to one line on the display. No effect if PC = 1.

Down arrow (on keypad) - Advances the PC one token. No effect if PC points to last token.

Delete - Eliminates the token pointed to by the PC.

Insert - Prompts the user for which type of token (single key command, string, or flow control construct) to insert and then for the specific command or string. The new token is inserted before the token pointed to by the PC. This permits multiple inserts to be performed without having to press Cursor Down after each insert.

Set PC - Prompts the user for a line number and moves the PC to that line.

```
:
:
     1    F
     4    <FOR Loop Variable 1 = to 99>
     11
     14   R
     17   "1"
     21   "DATA.001   "
     48   "DATA"
     55   C
PC>  58   F3
     61   W
     64   <Wait for user to press "Enter">
     66   N
     69   <NEXT>
     71   E
     74   <END of autosequence>
:
:
:
------------------EDITING AUTOSEQUENCE-- 75 used, 3925 free-

Set PC Insert before Delete
                          press space to return
```

Figure 5.11: A Typical Display Generated by Edit

5.6.4 Examples of Autosequence

Autosequences can be used to save keystrokes, or to perform calculations which normally cannot be done manually within WAVEFORM-CATALYST. Below is an example of each.

Example 1:

Suppose you have 10 data files on disk, named DATA.001 to DATA.010. You have just realized that something very curious can be seen in one particular spot of one of the files when expanded multiple times. You would like to look at this portion of each of the files. The thought of typing in each disk file name does not appeal to you, however. In this example, the autosequence feature can save you over 130 keystrokes!

Below is a listing of the procedure for letting CATALYST 'learn' the desired operations, assuming that the Main Display is on the screen showing the desired piece of one of the files. Between the L for enter Learn mode and the E for end Learn, the commands are shown in a format similar to that used by the autosequence editor.

```
F3              Autosequence control menu
L               Enter Learn mode
For             Loop Variable 1=1 to 10
                Space - return to Main Display
R               Read Data
'1'             Trace number
'DATA.001'      File name
'DATA'          Trace label - DATA.001 is now recalled to Trace 1
C               Connect dots
F3              Autosequence Learn menu
Wait            for user to press ENTER
4
Next
E               End Learn
```

Now press E for Edit, and replace the filename 'DATA.001' with 'DATA.LV1'. Press space to quit edit.

The above autosequence may now be run. It will recall each of the files 'DATA.001' through 'DATA.010', wait for user to press ENTER -- a single keystroke -- before proceeding with the next file. CATALYST remembers to show the same piece of each file.

Example 2:

A common calculation made from time data is integration to find the area under the curve between two points. Typically it is necessary to find the area under the curve but over a baseline. Because the autosequence feature has variables which may be operated upon, this can be done easily. The autosequence shown below calculates area under the curve between the primary and secondary cursors (inclusive), and uses the value at the primary cursor as the

threshold to be subtracted. The only assumptions are that the primary cursor is on the leftmost cell of interest and the secondary cursor is on the rightmost cell of interest when this autosequence is run.

Below is a listing of the procedures for letting CATALYST 'learn' this calculation. We will accumulate the result in V2:

F3	Autosequence control menu
L	Enter Learn mode
Assign V1 = dT	Save spacing between cursors
Assign V2 = dA + A	Get amplitude of second cursor - first point in sum
Assign V4 = A	Save primary cursor amplitude for threshold subtract
Second cursor	Set which cursor the left and right arrows move
Label "Moveleft"	
Cursor Left	Move the second cursor left (toward primary cursor)
Assign V3 = dA + A	Get amplitude of second cursor at this point
Assign V2 = V2 + V3	Accumulate
If dT > 0 go to Moveleft	Repeat until cursors are on same cell
Cursor right	Now primary and second cursor are one cell apart
Assign V2 = V2 * dT	Multiply sum of amplitudes by cell width in seconds
Assign V3 = V1 / dT	Number of cell widths, must add one to this
Assign V3 = V3 + 1	... because we are including both cursors
Assign V3 = V3 * V4	Threshold * number of cells summed
Assign V3 = V3 * dT	Area (Volts*seconds) correction for threshold
Assign V2 = V2 - V3	Subtract correction from area under curve
Print var 2	Print the result on the screen
Label "Moveright"	
Cursor Right	Move the second cursor back to start position
If dT !=V1 go to Moveright	
E	End Learn

5.6.5 Notes About Using Autosequence

1. Commands in CATALYST are context sensitive; that is, the same key has different meanings in different menus. Therefore, if you edit an autosequence, be sure not to insert or delete commands such that subsequent commands in the sequence will execute in the wrong menu. For example, inserting commands that change the menu from the Main menu to a module Setup menu and not inserting a space to change back. In this case, subsequent commands **N**, **G**, **E**, **P**, **R**, and **W**, which are valid in both menus, will have unexpected results. Execution will probably abort with an 'Expected Key' or 'Expected string' error message, indicating that a different type of command was expected next, at some point.

2. Labels save the 'context' information of the menu to which 'space' returns. The context saved is that which is current when the label is created. This context is restored when the label is encountered in Run mode. This ensures that execution will always be as desired when a learned label is passed or jumped to by an **I**. **WARNING:** The above is still true when a label is created by inserting it in the editor. If you must insert a label, be very sure that you came from the menu to which you want the next 'space' in the autosequence to return.

3. If you get an error message while in Learn mode that you do not expect to get at run time, then you must edit the autosequence to remove the extra commands necessary to correct the error. Otherwise the sequence of stored commands will be inappropriate at run time.

4. When inserting a string into an autosequence in the editor the maximum string length is 25 characters. This is more than adequate for all uses of strings except the user comment, which can be up to 160 characters. At run time, if CATALYST requests a string longer than the string in the autosequence, the first 25 characters are used.

5. Flow control tokens (shown by the editor in angle brackets) are logically tied to the preceding single key command (i.e., **F** always precedes a <FOR> token). Insert and Delete correctly handle these token pairs as single entities. Inserting a flow control construct will automatically cause insertion of an **F3** (Autosequence Learn/Run) command before it and a space (which returns to the menu in effect before the **F3**) after it, if they are necessary. If at the point where the insertion is desired the current menu would already be the Autosequence Learn/Run menu, then these extra tokens are not inserted.

6. If you get errors such as "More FOR's than NEXT's" when you end Learn mode or load an autosequence, the editor can be used to fix the sequence. If you get the errors ILLEGAL TOKEN TYPE or NOT LINKED after loading an autosequence, you have probably loaded a file that did not contain an autosequence. In the latter case, you still may edit it. Be warned, however. It may cause the PC

to hang up. If you see any of the following it is best to quit the Editor:

a. No 'PC>' marker on the display

b. Negative byte counts or byte counts greater than 4000

c. Several lines of 'Unknown token type', ... or blank lines.

7. The Help function, accessed by F1, and the User CAMAC menu List function generate displays for the user to read. Therefore, these functions will wait for user response before continuing, even in Run mode.

5.7 HARDCOPY OPTIONS

There are two different screen dumps provided by the system for an IBM Graphics (or equivalent) printer. The first screen dump provides a very fast hardcopy of the display. This printout has two limitations though, it plots on a 1:1 ratio (a somewhat elongated picture, Figure 5.12) and, since it is produced by pressing P when the main display is active, it cannot be used if a menu is active. The second screen dump (Figure 5.13) is printed when you hold down the SHIFT key and press the one labeled PRTSC. This one takes twice as long as the first one, but provides a picture with the same aspect ratio as the screen. It can be accessed from any menu in the system. Note - SHIFT + PRTSC does not work if CATALYST is in color.

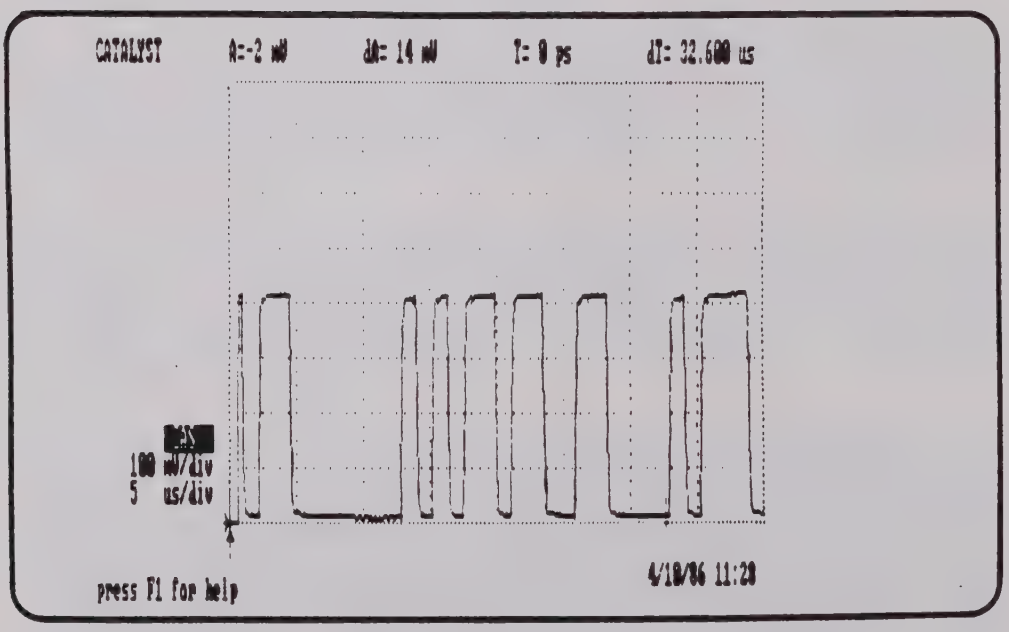


Figure 5.12 Fast Screen Dump

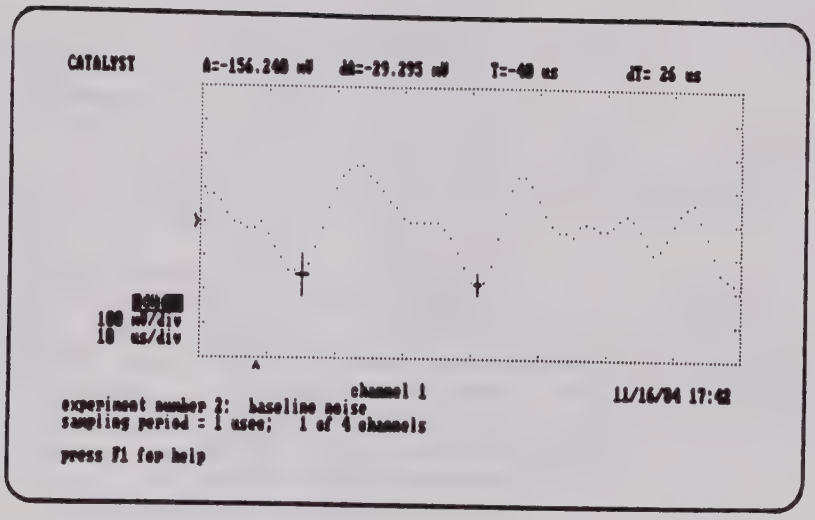


Figure 5.13 Image Screen Dump

5.8 DISK UTILITIES

This section describes various commands that you can use to transfer data to and from the disk drives.

Read data - Reads a file containing previously stored data into the system. You will be asked to supply the filename which may include the drive path and a file extension. (e.g. C: DATA ADC8.1) If the drive or path is not specified, the one specified in DEFAULT.CAT is used. You will also be asked for the trace number corresponding to the location of data display. Once the trace is assigned, the data is treated as if it were coming from a module. In other words, you can page, expand, or perform other operations on the data. When the data is read from the disk, the comment that was stored along with it is printed on the screen over the current comment. When the screen is redrawn for any reason, the current comment is redisplayed.

Write data - Write allows you to save the data from a module onto disk for later reference. Along with the data, CATALYST stores all the parameters it can read from the module plus any other information needed to redisplay the data when it is recalled. When you request that data be saved, the system asks you to supply a valid DOS filename, ([d:][\PATH\]filename[.ext]). If a drive or path is not specified, the one in DEFAULT.CAT (see Section 6) is used. After the name is entered, you must specify the module or trace label from which the data should be read and saved.

F2 - Pressing the second function key gives you a menu of disk management commands which let you list the disk directory, erase a file, or rename one. Each of these commands are discussed below. Since it is a function key, this command may be accessed at any time.

Change Drive - Allows you to change the drive and/or path used for data storage (originally specified in DEFAULT.CAT). You must type drive letter, ":", path name (up to 9 characters) "\", and then ENTER.

Directory - asks for which drive or path (if any) you

want displayed and then prints it on the screen. The directory will remain on the screen until you leave this menu by pressing the space bar.

Erase - asks for the name of the file to be deleted. The filename may include the drive, path, and file extension if any.

Rename - first asks you for the existing filename, and then for the new name you wish to give it. For both names you must give valid DOS filenames (see DOS manual for details).

Set write format - For each possible storage format (CATALYST, ASYST, ILS) the user is prompted for a Yes/No answer specifying if disk storage in that format is desired. The current setting is displayed. Any response other than **y** or **Y** is interpreted as "no". Note: these settings may be specified in the file DEFAULT.CAT; see Section 6.

5.9 Miscellaneous Commands

Trace Restore - Reads and displays the data from each module assigned to a trace. This allows you to redisplay data from a module that had not been assigned to a trace when the trigger occurred.

F7 - Causes a calibration cycle on all instruments (except 6880's) which require it.

F8 - Asks if you want to perform a DC or full calibation on all 6880's present. DC calibration should be performed about 10 minutes after the system is powered on and any time there is a temperature change of more than a few degrees in the operating environment. Full calibration may take several minutes to complete and should be performed about once a day.

F10 - Pressing this function key causes you to exit CATALYST. Control is transferred to DOS - the operating system.

0 - This command lets you reset the event counter. The event counter is incremented once per event where an event includes

arming the modules and reading them in regardless of how many modules are active. The 'auto store' feature uses the counter along with the modules' name to generate the filename for the data. In this way a unique name is created for each file and all data taken at the same time can be easily identified. For example, if you had two modules TR1 and TR2 enabled for auto store, after the first sweep the files TR1.1 and TR2.1 would be on the disk. After the second sweep TR1.2 and TR2.2 would also be there. If the counter is not reset, it will roll over to zero after 999 events. This is many more files than can fit in one disk directory.

5.10 WAVEFORM LIMIT TESTING

CATALYST can automatically compare acquired data to a predefined set of limits and report the number of times the data exceeds the limits. It provides the ability to store and/or print the data if an error occurs. (See Figure 5.14). To access this feature press the Yes/No test key on the main menu and the following menu will be displayed:

Action on error - allows you to specify what you want the system to do when it finds data outside of the specified range. You are prompted for each possible action - you can choose any or all of them. The choices are: Store the data to disk, Print the screen, Print only the number of errors and the time and date, and Freeze the trigger.

Create - lets you create the limits about the reference. You are first prompted to specify which trace (1 or 3) to use as the reference. Next, you set the upper limit by using the up and right arrow on the keypad (the down and left arrows allow you to back up if you need to). The difference in time and amplitude from the reference is printed on the side. When you are done setting the upper limit, press the spacebar. Now you can set the lower limit with the left and down arrows (use the up and right arrows for editing). When the lower limit is where you want it, press the spacebar again.

Read - lets you read a previously stored envelope from disk to either trace 2 or 4.

Write - allows you to store an envelope on disk for later use.

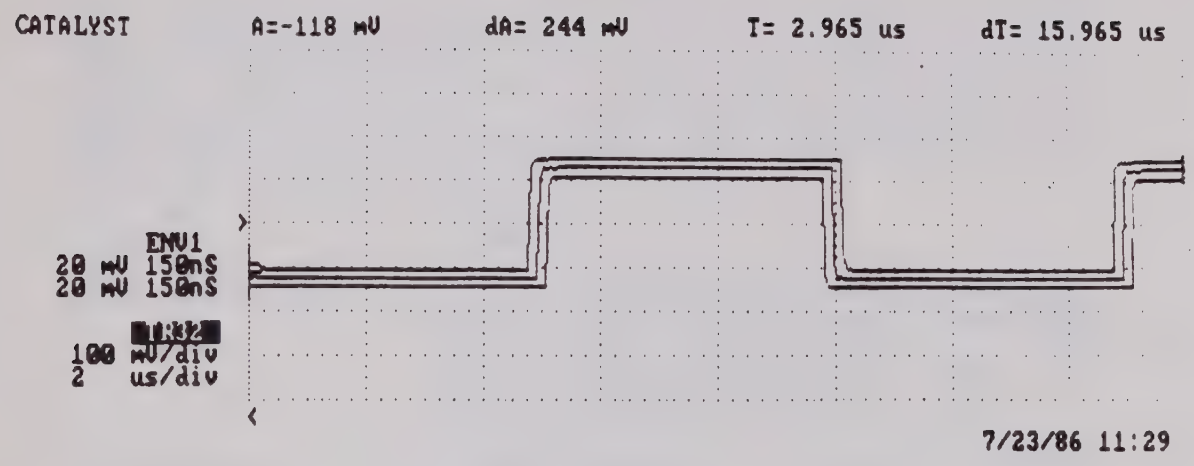


Figure 5.14 Waveform Limit Testing

Example Procedure:

1. Put the signal to be used as a reference to create envelope on either trace 1 or 3. You can use any valid trace (i.e., from disk or live data). Catalyst allows you to test up to two waveforms at one time. The waveforms must be placed on trace 1 and 3 and their corresponding envelopes (limits) on trace 2 and 4, respectively.

2. Press Y for yes/no test.

3. Create the limits you desire.

4. Select what action should be performed when an error occurs.

5. When an envelope is present, all newly acquired data is automatically compared to the limits and if an error is found the specified action is performed. (Note that it is recommended that the anti-aliasing display algorithm and not 'quick draw' be used when running the limit tests).

6. To stop the testing, simply delete the limit envelope by pressing the number corresponding to its trace (2 or 4) and then ENTER.

6. CHANGING PARAMETER DEFAULT CONDITIONS

On the system disk is a file named "DEFAULT.CAT". This file contains the initial settings for various system parameters (see figure 6.1). By changing these values, you can customize the system so that it starts up the way you want it to. To change these values you must edit it with an editor. The DOS editor is called EDLIN and is described fully in the DOS manual. The purpose of each value is described in detail in this section.

ITEM	PURPOSE
8901 compatibility	allows CATALYST to be backwards compatible with LeCroy 8901 controllers. If any 8901's are present, this should be 0. If only 8901A's and/or 6010's and 9400's, this switch should be set to 1.
grid format	selects single, dual, or quad initial grid
grid status	specifies if grid is initially on or off
trace 1	specifies which module, if any, will have its data displayed on trace 1.
trace 2	specifies which module, if any, will have its data displayed on trace 2.
trace 3	specifies which module, if any, will have its data displayed on trace 3.
trace 4	specifies which module, if any, will have its data displayed on trace 4.
points	specifies whether or not the points of the trace should be connected with line segments.
display	specifies which binning algorithm should be used for display processing
system mode	If the first character on this line is 0 or blank, CATALYST powers up in interactive (normal) mode. Otherwise this line is assumed to contain a path name of an autosequence file to be automatically loaded and run at power up. The path may be up to 24 characters.
trigger mode	selects which of the possible trigger modes the system should start up in.
disk drive and write format	specifies the disk drive and path name to be used for storing data. When storing data,

CATALYST first determines if the drive has enough room to store the entire file. If there is not enough room, CATALYST will automatically switch to the next drive (e.g.. from B to C). If this drive does not exist you will be prompted with the message "DISK FULL - ENTER NEW DRIVE [PATH]" at which time you can put in a new disk and enter the drive and path name. You can also change the drive and/or path with the Change drive command. Note that up to 12 characters may be used to specify the default drive and path, and the last non-blank character MUST be a backslash (e.g. B:\DATA\ or C:\).

After one (or more) blanks, the characters 'C', 'A', 'I' are searched for, up to a '#' or the end of the line, or up to the 20th column, whichever comes first. If none are found, CATALYST will store data in the format specified in Appendix A. The same result may be obtained by using only the letter 'C'. 'A' specifies storage in ASYST's format (ASYST - MacMillan Software Company) and 'I' specifies storage in ILS's format (ILS-PC - Signal Technology, Inc.). Note: if 'A' or 'I' or both, but NOT 'C' are specified, CATALYST will write a file in its own format, convert it to the desired format(s) and then delete the CATALYST format file.

Examples:
```
C:\    # store in CATALYST format
C:\  C # store in CATALYST format
C:\ CA # store in CATALYST and ASYST format.
```

Please see Appendix B for more information on storing data in ASYST or ILS format.

CAMAC registers

this specifies the name of the file containing the startup value for the User CAMAC "registers". This file is an ASCII file which allows it to be modified with a text editor (e.g. EDLIN). The format of the data for each of the 9 registers is:

```
value for crate for register n
value for Slot for register n
value for ADDR for register n
value for FUNC for register n
value for DATA for register n
```

module type, setup data, and Help information

Three files are needed for each module type in the system. The first file specifies information needed (such as name, allowable values for parameters, etc.) about the modules. This file can be changed only with the DEFMOD Program (see CATALYST Function Library Manual for details). The second file contains the default parameters for each module of this type. To change the module parameters you must start CATALYST, go into the setup menu, and enter your choice of values in the parameter table. Then you must write the table to disk. (e.g. store setup for TR8837 in SD8837.CAT). Subsequently, whenever you run CATALYST, it gets the default parameter filename (i.e. SD8837.CAT) from DEFAULT.CAT and programs the modules accordingly. The third file contains Help information about the particular module type. This information is displayed when Help is requested from the setup menu. All of these files should be on the same drive and directory as the CATALYST Program file.

```
1          #no 8901's present
1          #grid format (single=1; dual=2; quad=4)
1          #grid status (on=1; off=0)
TR28       #trace 1
           #trace 2
           #trace 3
           #trace 4
           #un/connect points (=0/1)
           #display processing option (fast = 1; min/max=2)
           # up to 24 char path to auto-exec autosequence, or 0 or blank
           #sweep mode (disabled=0; frozen=1; auto=2; normal=3; single=4)
C:DATA/\      #default drive, path name for data storage (MUST end with \)
sdcamac.cat   #user camac command registers
mtd8837.cat   #module type 1 data
sd8837.cat    #setup data for type 1
dc8837.hlp    #help file for type 1
mtd8818.cat   #module type 2 data
sd8818.cat    #setup data for type 2
dc8818.hlp    #help file for type 2
mtd8828.cat   #module type 3 data
sd8828.cat    #setup data for type 3
dc8828.hlp    #help file for type 3
mtd6880.cat   #module type 4 data
sd6880.cat    #setup data for type 4
dc6880.hlp    #help file for type 4
mtd6103.cat   #module type 5 data
sd6103.cat    #setup data for type 5
dc6103.hlp    #help file for type 5
mtd8212.cat   #module type 6 data
sd8212.cat    #setup data for type 6
dc8212.hlp    #help file for type 6
mtd8210.cat   #module type 7 data
sd8210.cat    #setup data for type 7
dc8210.hlp    #help file for type 7
mtd8100.cat   #module type 8 data
sd8100.cat    #setup data for type 8
dc8100.hlp    #help file for type 8
mtd9400.cat   #module type 9 data
sd9400.cat    #setup data for type 9
dc9400.hlp    #help file for type 9
mtd4222.cat   #module type 10 data
sd4222.cat    #setup data for type 10
dc4222.hlp    #help file for type 10
```

Figure 6.1 Contents of DEFAULT.CAT

APPENDIX A
Format of Stored Data

The following is a description of what is contained in the data file stored by WAVEFORM CATALYST.

In addition to the format described below, CATALYST can also create files in formats suitable for ASYST and ILS processing packages. Please refer to each package's documentation for information on the file structure each requires.

A1. FORTRAN Unformatted Sequential Files

CATALYST stores data as an "unformatted", sequential file. FORTRAN unformatted files contain record delimiters in addition to the desired data. If the open statement used to read a FORTRAN unformatted file specifies "form='UNFORMATTED'" then the record markers are properly used by FORTRAN and are invisible to the user. Each READ statement uses up one logical record from the file, regardless of how much data is actually read. It is therefore necessary to READ everything desired from each record in one READ statement; a loop to read bytes will read the first byte of successive logical records. The logical records in a CATALYST data file are:

1. the header - 17 2-byte integers
2. the user comment, terminated by '$' - 161 characters
3. through last successive data blocks as described by the file header.

See the example FORTRAN program at the end of this appendix for an example of the proper way to read a CATALYST data file.

If a FORTRAN unformatted file is opened as a BINARY file by FORTRAN (or is opened by any other language) then the record delimiters will not have their intended effect and will be visible to the program reading the file. Below are details of the structure of a FORTRAN unformatted sequential file.

The first byte in any FORTRAN unformatted sequential file is the value 75. All the rest of the file is made up of physical records of up to 128 bytes of data. Each physical record is preceded and followed by a byte containing the number of data bytes in that physical record. There is one special case: If a logical record (that is, the data written by one WRITE statement) contains more than 128 bytes, then more than one physical record is required. In this case, one or more 128-byte physical records are written, preceded and followed by the value 129; the final physical record is preceded and followed by a byte containing its actual length.

See the example BASIC program at the end of this appendix.

A.2 CATALYST file header

DATA BYTES	USE
1-2	Block size - number of data values per block
3-4	Number of bits per data word. Each word occupies an integer number of bytes. For example, 12-bit data occupies 2 bytes per value.
5-8	Period of trace in units of 0.1 nsec per point. For time data, see also bytes 29-30.
9-10	Offset (code value corresponding to zero volts)
11-14	Point where trigger occurred
15-18	Amplitude in micro units per code. For time data, units are Volts. For histogram or point values, see also bytes 29-32.
19-20	Offset (in bytes) to actual start of data
21-22	Total number of data blocks (length of data divided by block size)
23-24	Type of data: 0 = single channel time vs amplitude, 1 = multichannel time vs amplitude, 2 = histogram, 3 = point values, 4 = module specific settings. Note: averaged time data also is stored as type 0.
25-34	Type dependent data.
	If type of data is 0 or 1:
	25-26 blocks per channel
	27-28 start channel (type 1 only)
	29-30 Exponent for sampling period. Divide the value from bytes 5-8 by ten raised to this value to obtain a value where 1 = 0.1 nsec.
	If type of data is 2 or 3:
	29-30 Exponent for amplitude. Divide the value from bytes 15-18 by ten raised to this value to obtain a value where 1 = 1 micro unit.
	31-32 Character for measurand units. 'S' for seconds, 'C' for coulombs, 'V' for volts.
	If type of data is 2 (in addition to above):
	25-28 lovest amplitude code included in this histogram.
	33-34 number of codes per "bin".

Below is an example of what the beginning of a CATALYST data file looks like when it is displayed by the DOS debugger. An explanation follows. All values are in base 16. Note that where a value is made up of two or more bytes, the first byte is the least significant and the last byte of the value is the most significant.

```
Offset    Data

00        4B 22 00 20 08 00 32 00-00 00 80 00 00 00 00 00
10        D0 07 00 00 01 00 04 00-00 00 04 00 00 00 00 00
20        00 00 00 00 22 81  ...  (user comment)
A0                    ...  81 21- ...  (user comment)
C0                              -24 21 81 ...  (data)

off-
```

set	value	Explanation
00	4B	FORTRAN unformatted sequential file
01	22	Length of the physical record to follow is 34 bytes. This one physical record corresponds to a logical record. The 34 bytes are the CATALYST file header, which is explained in detail below.
24	22	Length of the preceding physical record.
25	81	Special value 129 indicates that the length of this physical record is 128 bytes and that it is the first of at least two physical records in one logical record. This physical record contains the beginning of the 161-byte user comment.
A6	81	End of the preceding physical record.
A7	21	Length of the physical record to follow. This is the last physical record in the logical record containing the user comment, it contains the last 33 bytes.
C9	21	Length of the preceding physical record.
		Data records follow, 1 block per logical record. The number of bytes in each data block is defined in CATALYST's file header.

The 34 byte CATALYST header is interpreted as follows:

00 20	# of values per block of data = 8192
08 00	Width of each data value = 8 bits
32 00 00 00	Sampling period in units of 0.1 nsec = 5.0 nsec/pt
80 00	Offset (code corresponding to zero volts) = 128
00 00 00 00	Point corresponding to trigger time = first sample
D0 07 00 00	Amplitude in microvolts/code = 2.0 mV/code
01 00	Start byte (1 means skip one byte before 1st value)
04 00	Total number of blocks of data = 4
00 00	Type of data = single channel time vs amplitude
04 00	Number of blocks per channel
00 00	Unused for time data
00 00	Exponent for sampling period
00 00	Unused for time data
00 00	Unused for time data

A.3 Sample FORTRAN Program

The sample FORTRAN program segment below can be used to read a block of data from the specified file. This segment is meant only as an example of how to read in the data and is not a complete program.

```
      PROGRAM SAMPLE
C
C     VARIABLE DEFINITIONS
C
C     HEADER - temporary buffer          LENGTH - number of data words
C     WIDTH - # of bits/word                  PERIOD - period of trace
C     OFFSET - zero volt offset                   TRIG - trigger point
C     AMPL - amplitude                              START - start byte
C     BLKCNT - # of blocks                        TYPE - type of data
C     BLKCHN - blocks/channel                CHNLST - starting channel
C     DATBUF - array for reading one block of data
C
      INTEGER*2 SDFILE, HEADER(17), LENGTH, TYPE, START, BLKCNT
      INTEGER*2 LEN, BLOCK, I, DATBUF(4100)
      INTEGER*2 BLKCHN, CHNLST, PTR
      INTEGER*4 TRIG, PERIOD, AMPL
      CHARACTER*1 TEXT(161)
      CHARACTER*14 FNAME
      DATA SDFILE/5/
C
C     prompt operator for filename
      WRITE(*,'(A\)')' ENTER NAME OF FILE'
      READ(*,'(A)')FNAME
C     read in header information
      OPEN(SDFILE, FILE=FNAME, STATUS='OLD', FORM='UNFORMATTED')
      READ(SDFILE)(HEADER(I), I = 1, 17)
      READ (SDFILE) (TEXT(I), I = 1, 161)
      LENGTH = HEADER(1)
      WIDTH = HEADER(2)
      PERIOD = HEADER(4) * 65536
      PERIOD = PERIOD + HEADER(3)
      OFFSET = HEADER(5)
      TRIG = HEADER(7) * 65536
      TRIG = HEADER(6) + TRIG
      AMPL = HEADER(9) * 65536
      AMPL = HEADER(8) + AMPL
      START = HEADER(10)
      BLKCNT = HEADER(11)
      TYPE = HEADER(12)
      BLKCHN = HEADER(13)
      CHNLST = HEADER(14)
C     ask operator for block number desired and read it in
      WRITE(*,'(A\)')' ENTER BLOCK TO BE READ'
      READ(*,'(I2)')PTR
      BLOCK = 0
      LEN = (WIDTH+7)/8
      LEN = (LEN*LENGTH+1)/2
10    BLOCK = BLOCK + 1
      READ(SDFILE)(DATBUF(I), I = 1,LEN)
```

```
      IF (BLOCK.NE.PTR) GOTO 10
      CLOSE(SDFILE)
C     at this point, the selected block of data is contained in
C     DATBUF and the module parameters are in their corresponding
C     variables you can add a processing or print routine here.
      END
```

A.4 Sample BASIC Program

Below is a sample BASIC Program segment which reads the data from disk into an integer array. (Note: this is written for 8-bit data words and up to 16K words.)

```
10   DEFINT A-Z
20   OPTION BASE 1
30   DIM IDATA (2,8192)
40   INPUT "ENTER NAME OF DATA FILE"; FILENAME$
50   OPEN FILENAME$ FOR INPUT AS #1
60   V=ASC(INPUT$(1,#1))
70   IF V < > 75 GOTO 590
80   COUNT = ASC(INPUT$(1,#1))
90   LENGTH = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
100  PRINT "NUMBER OF BYTES PER BLOCK",LENGTH
110  WWIDTH = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
120  PRINT "BITS PER WORD", WWIDTH
130  PERLOW = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
140  PERHI = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
150  PERIOD! = PERLOW + PERHI*65536
160  PRINT "PERIOD IN .1 NANOSECONDS",PERIOD!
170  OFFSET + ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
180  PRINT "ZERO VOLTS = A VALUE OF", OFFSET
190  TRIGLOW = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
200  TRIGHI = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
210  TRIGOFF! = TRIGLOW + TRIGHI*65536
220  PRINT "TRIGGER OCCURRED AT",TRIGOFF!
230  AMPLOW = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
240  AMPHI = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
250  AMPL! = AMPLOW + AMPHI*65536
260  PRINT "AMPLITUDE (MICROVOLTS/CODE)", AMPL!
270  START = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
280  PRINT "START ON BYTE",START
290  BLKCNT = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
300  PRINT "NUMBER OF BLOCKS", BLKCNT
310  TYPE = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
320  IF TYPE = 0 THEN PRINT "TIME VS AMPLITUDE DATA"
330  IF TYPE = 1 THEN PRINT "MULTICHANNEL DATA"
340  IF TYPE = 4 THEN PRINT "POINT DATA"
350  TDEPEND1 = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
360  TDEPEND2 = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
370  TDEPEND3 = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
380  TDEPEND4 = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
390  TDEPEND5 = ASC(INPUT$(1,#1)) + ASC(INPUT$(1,#1))*256
400  PRINT TDEPEND1,TDEPEND2,TDEPEND3,TDEPEND4,TDEPEND5
410  COUNT = ASC(INPUT$(1,#1))
420  COUNT = ASC(INPUT$(1,#1))
430  MSG1$ = INPUT$(128,#1)
440  COUNT = ASC(INPUT$(1,#1))
450  COUNT = ASC(INPUT$(1,#1))
460  MSG2$ = INPUT$(33,#1)
470  PRINT MSG1$,MSG2$
480  COUNT = ASC(INPUT$(1,#1))
```

```
490  FOR BLOCK = 1 TO BLKCNT
500   FOR INDEX = 1 TO LENGTH STEP 128
510     COUNT = ASC(INPUT$(1,#1))
520     FOR PTR = 0 TO COUNT -2
530     IDATA(BLOCK,PTR+INDEX) = ASC(INPUT$(1,#1))
540    NEXT PTR
550    COUNT = ASC(INPUT$(1,#1))
560  NEXT INDEX
570  NEXT BLOCK
580  GOTO 600
590  PRINT "INVALID FILE TYPE"
600  END
```

APPENDIX B

Interface to Signal Processing Software

CATALYST can now store data on disk in three formats: its own, a format suitable for ASYST, and a format suitable for ILS.

In order to process data stored on disk in CATALYST's file format only, two utility programs, CATILS and CATASYST are supplied on the CATALYST Boot Disk. CATILS converts CATALYST data files to a format usable by the ILS-PC signal processing programs. ILS-PC permits flexible display and basic processing of the data (including FFT). ILS-PC2 permits more sophisticated processing. CATASYST converts CATALYST data files to a format usable by the ASYST signal processing programs. ASYST module 1 permits flexible display and basic processing of the data. ASYST module 2 permits more sophisticated processing including FFT, IFFT, etc.

ILS is a product of Signal Technologies, Inc. ASYST is distributed by MacMillan Software, Inc. LeCroy does not warrant either of these products in any way. The existence of CATILS and CATASYST should not be viewed as a recommendation by LeCroy of either package's performance or fitness for any particular purpose.

The use of these utility programs is described below:

B.1 CATILS - CATALYST to ILS Interface

B.1.1. Running CATILS from a Floppy Disk Based System

CATILS creates its output file(s) in the "current directory". If your PC does not have a hard disk, you will probably want to read CATALYST data files from one floppy and have CATILS write its output to a second floppy. This is accomplished as follows:

1. Use DOS commands to set your "current directory" to where you want the output files to reside. For example, if you want the output created on a (formatted) disk in drive B, in the root director, type B:

2. Execute the programs by giving DOS the path to where the executable file resides. For example, the program is supplied in the root directory of your CATALYST disk. If this disk is in drive A, type A:\CATILS

3. There are three possibilities for location of the CATALYST data files: Either they are on the disk in drive A, on the disk in drive B (the current drive), or they are on a third floppy not currently in either drive.

   If the CATALYST data file is on the disk in drive A, give the pathname when prompted for the name of the CATALYST file to convert.

If the CATALYST data file is on the disk in the current drive, you can just give the file name when prompted.

If the CATALYST data file is on a third disk, then as soon as you see the program's first prompt, "Name of CATALYST file...", you may remove the disk currently in drive A and insert the disk with the data into drive A. Then type in the pathname of the desired file.

B.1.2 Running CATILS on Hard Disk Based Systems

You have probably already created a large directory containing all the ILS programs and data files. CATILS.EXE should be copied into this directory. If this directory is the current directory, then simply type CATILS to execute the program and give the path to the input file when prompted. The output files are always created in the current directory.

B.1.3 Examples

Figures B.1 and B.2 show what would appear on a PC's monitor after running CATILS. Both examples were generated on an IBM PC-AT. Probable differences for a floppy-based system will be noted below.

Both examples begin with the "current directory" set to where the output data files will reside. The executable file CATILS.EXE also resides in this directory. The first line of both examples therefore shows the program being invoked by typing CATILS.

The program prompts, "Name of CATALYST file to convert to ILS Sampled Data File?". If the desired file is in the current directory you may simply give its name, as in Figure 1. If the desired file is not in the current directory you must give the pathname, as in Figure B.2 (the input file name is "sin4ch" which is in directory \cat\prog\data on the current disk, drive C.). Users of floppy-based systems may have to include a drive specifier, such as "A:", in the pathname if the input file is on a different drive than the current directory; see B.1.1 for details.

ILS data files have names made up of two alpha characters followed by up to four digits (with no leading zeros). CATILS will prompt you for the two characters and the number separately, as shown in the figures. As soon as you enter the numerical part of the ILS file name CATILS will create the (first) output file and begin converting the CATALYST data file to ILS's format.

Nothing further is entered by the user. In particular, note that the you do not need to know anything about the data file except its name. All other information (such as number of channels, number of bits per A/D word, amplitude scale factors, etc.) is obtained from the CATALYST file header.

If the CATALYST data file contains single channel data, CATILS will create one output file with the name specified by the user, as shown

in Figure B.1. If the CATALYST data file contains multi-channel data, CATILS will create one output file for each channel of data, as shown in Figure B.2. CATILS adds the channel number to the numerical part of the ILS file name requested by the user when the input file contains multi-channel data. If the requested numerical part of the file name had one or two trailing zeros, then channel number of each data file will be obvious from the names of the data files.

CATILS will tell you as it finishes creating each output file. Figure B.1 shows one such message because the input was single-channel data. Figure B.2 shows four such messages because the input file had 4-channel data. If CATILS created multiple output files (due to multichannel input) it will also display "Created one output file per input channel, filenames are" and a summary of the filenames. This is useful when the input file contains data from more than a dozen channels, because some of the individual messages "Finished writing file <filename>" will roll off the display.

Finally, CATILS displays a message containing how to tell ILS about the A/D which created this data. This is necessary because ILS does not associate this information with each data file, but treats it as global information about all data files. The ILS default is for a 12-bit A/D converter.

The message "Normal termination" indicates that CATILS has ended of its own accord instead of by an error from DOS. The DOS prompt will appear when the program terminates.

```
C>catils
Name of CATALYST file to convert to ILS Sampled Data File?
spulse.cln

Please enter the two character prefix for ILS filename?
sp

Please enter numerical part of ILS filename <0 to 9999?
500
Finished write file sp500

You  must tell ILS the #bits and mid-value of the A/D converter which
collected these data.  The correct command is:

       ILS C,,,, 8, 128

assuming that an ILS common file already exists.
Normal termination

C>
```

Figure B.1

```
C>catils
Name of CATALYST file to convert to ILS Sampled Data File?
\cat\prog\data\sin4ch

Please enter two character prefix for ILS filename?
si

Please enter numerical part of ILS filename <0 to 9999>?
400
Finished writing file si401
Finished writing file si402
Finished writing file si403
Finished writing file si404

Created one output file per input channel, filenames are si 401
through 404.  ILS can multiplex, see MDX command.

You must tell ILS the #bits and mid-value of the A/D converter which
collected these data.  The correct command is:

      ILS C,,,, 8, 128

assuming that an ILS common file already exists.
Normal termination
```

Figure B.2

B.2 CATASYST - CATALYST to ASYST Interface

B.2.1 Running CATASYST from a Floppy Disk Based System

CATASYST creates its output file(s) in the "current directory". If your PC does not have a hard disk, you will probably want to read WAVEFORM-CATALYST data files from one floppy and have CATASYST write its output to a second floppy. This is accomplished as follows:

1. Use DOS commands to set your "current directory" to where you want the output files to reside. For example, if you want the output created on a (formatted) disk in drive B, in the root directory, type: B:

2. Execute the program by giving DOS the path to the executable file. For example, if the program is in the root directory of the disk in drive A, type A:CATASYST

3. There are three possibilities for location of the CATALYST data files: Either they are on the disk in drive A, on the disk in drive B (the current drive), or they are on a third floppy not currently in either drive.

   If the CATALYST data file is on the disk in drive A, give the pathname when prompted for the name of the CATALYST file to convert.

   If the CATALYST data file is on the disk in drive B, you can simply give the file name when prompted.

   If the CATALYST data file is on a third disk, then as soon as you see the program's first prompt, "Name of CATALYST file...", you may remove the disk currently in drive A and insert the disk with the data into drive A. Then type in the pathname of the desired file.

B.2.2 Running CATASYST on Hard Disk Based Systems

You have probably already created a large directory containing all the ASYST program and data files. CATASYST.EXE should be copied into this directory. If this directory is the current directory, then simply type CATASYST to execute the program and give the path to the input file when prompted. The output files are always created in the current directory.

B.2.3 Examples

Figures B.3 and B.4 show what would appear on a PC's monitor after running CATASYST. Both examples were generated on an IBM PC-AT. Probable differences for a floppy based system will be noted below.

Both examples begin with the "current directory" set to the place that output data files will reside. The executable file CATASYST.EXE also resides in this directory. The first line of both examples

therefore shows the program being invoked by typing CATASYST.

The program prompts, "Name of CATALYST file to convert to ASYST Sampled Data File?". If the desired file is in the current directory you may simply give its name, as in Figure B.3. If the desired file is not in the current directory you must give the pathname, as in Figure B.4. (In Figure B.4, the input file name is "sin4ch" which is in directory \cat\prog\data on the current disk, drive C.) Users of floppy based systems may have to include a drive specifier, such as "A:", in the pathname if the input file is on a different drive than the current directory; see B.2.1 for details.

ASYST data files normally have any legal DOS filename, but CATASYST will ask you for a filename without an extension. The reason for this will be explained below. As soon as you enter the ASYST file name CATASYST will create the (first) output file and begin converting the CATALYST data file to ASYST's format.

Nothing further is entered by the user. In particular, note that the user does not need to know anything about the data file except its name. All other information (such as number of channels, number of bits per A/D word, amplitude scale factors, etc.) is obtained from the CATALYST file header.

If the CATALYST data file contains single channel data then CATASYST will create one output file with the name specified by the user, as shown in Figure B.3. If the CATALYST data file contains multichannel data, CATASYST will create one output file for each channel of data. In this case CATASYST creates an extension to the file name requested by the user to show which channels' data each output file contains, as shown in Figure B.4.

CATASYST will tell you as it finishes creating each output file. Figure B.3 shows one such message because the input was single-channel data. Figure B.4 shows four such messages because the input file had 4-channel data. If CATASYST created multiple output files (due to multichannel input) it will also display "Created one output file per input channel, file names are" and a summary of the filenames. This is useful when the input file contains data from more than a dozen channels, because some of the individual messages "Finished writing file <filename>" will roll off the display.

Finally, CATASYST displays a message explaining how to tell ASYST about the frequency at which this data was sampled. This is necessary because ASYST modules 1 and 2 do not know about sampled data, only about arrays of numbers.

The message "Normal termination" indicates that CATASYST has ended of its own accord instead of by an error from DOS. The DOS prompt will appear when the program terminates.

B.2.4 ASYST Data Files as Produced by CATASYST

The general format of ASYST data files is explained in ASYST's module 1 manual, Appendix B.

CATASYST produces ASYST files with only one type of subfile, repeated as many times as necessary to hold the required data. The subfiles are one-dimensional arrays of single precision real numbers. The size of that dimension is the number of values stored in one CATALYST data record, which is typically 8192 for byte data (as from TR8828's) or 4096 for wider data, which CATALYST stores as 16-bit integers. As noted in the ASYST manual, the most significant limitation on the size of ASYST files is that the file template must not exceed 1024 bytes. This permits 494 subfiles in this case (because there is only one subfile template, and each repetition uses 2 bytes). ASYST files produced by CATASYST can therefore contain more 2 million values if each subfile holds 4096 values, or more than 4 million values if each subfile holds 8192 values.

CATASYST produces ASYST files with four (64 character) comment lines each. The first two-and-a-half lines are used to hold the 160-character CATALYST user comment from the input file. CATASYST fills the second half of the third comment with the channel number (only if multichannel input) and the name of the CATALYST data file from which this data was taken. CATASYST also fills most of the fourth line with the sampling interval in seconds and frequency in Hertz.

```
A>catasyst
Name of CATALYST file to convert to ASYST file(s)?
sine.10

Please enter eight character ASYST file name (no extension)?
sine
Finished writing file sine

ASYST knows about arrays, not sampled data.  The 4th comment shows
the sampling period.  This can be used to make a ramp so XY.AUTO.PLOT
will correctly annotate time.

The output data is real, in units of Volts.
Normal termination

A>
```

Figure B.3

```
C>catasyst
Name of CATALYST file to convert to ASYST file(s)?
\cat\prog\data\sin4ch

Please enter eight character ASYST file name (no extension)?
sin4chan
Finished writing file sin4chan.1
Finished writing file sin4chan.2
Finished writing file sin4chan.3
Finished writing file sin4chan.4

Created  one  output file per input channel, filenames are sin4chan.1
through  4.

ASYST  knows  about  arrays, not sampled data.  The 4th comment shows
the sampling period.  This can be used to make a ramp so XY.AUTO.PLOT
will correctly annotate time.

The output data is real, in units of Volts.
Normal termination

C>
```

Figure B.4

B.2.5 USEFUL ASYST WORDS FOR FFT'S

ASYST's FFT does the mathematical procedure known as a Fast Fourier transform. There are several things the user will probably want to do in addition to the actual FFT. They are:

1. Weight the time data to minimize leakage due to non cell-centered signals.

2. Scale the FFT result down by N/2 (necessary because the energy seen at each frequency of a continuous signal is doubled each time the length of the input is doubled. The divide-by-two arises because the energy is split between the positive and negative frequency parts) so that a 1 V center-to-peak sine wave produces a 1 V peak in frequency domain.

3. Display the result as magnitude with scaled and labeled axis.

The file FFT1024.LOG contains ASYST "Colon Definitions" which define three new ASYST "words" to perform the functions discussed above. This file is on the CATALYST SUPPLEMENTAL PROGRAM disk.

Books have been written about exactly what an FFT does. The user needs to be aware of some possibly unexpected limitations:

1. The shorter the input record, the worse the frequency resolution. Multiple copies of the same short input record do not improve frequency resolution.

2. The FFT looks for those frequencies which have an exact integer number of cycles in the input record. These are called "cell centered" frequencies. The energy of non-cell-centered signals is divided between multiple cells, causing the peak in the frequency domain to be reduced in amplitude. Weighting reduces, but does not eliminate, this effect.

3. Weighting is useful only for continuous signals with discrete frequency components. Weighting should not be applied to transients, as the resulting spectrum will depend on the position of the transient in the input record. Weighting should not be applied to noise-like (continuous but broadband) signals, because the amplitude in the frequency domain will be too large.

4. If a 1 V peak sine wave produces a 1 V peak in the output of the FFT, then a 1 V DC offset will produce a 2 V peak at zero Hertz in the output of the FFT. The words in the file FFT1024.LOG make no attempt to correct for this. Simply remember that the point on the left edge of the graticule (zero Hertz) is too high; and that weighting may spread this artificial peak by a couple of cells.

B.2.5.1 Step-by-Step Instructions for your First FFT

We will assume that you have followed the instructions in Section 2 to produce an ASYST format file from a CATALYST data file. For illustration we will use the names MYFILE and MYARRAY as the names of the ASYST format file and the named array we will create in ASYST.

Here is one of the correct sequences of commands to open the file "MYFILE", read in a large chunk of it, and FFT it.

ASYST	ASYST logo comes up. Master floppy must be in one of the drives. ASYST says "type any key to continue"
(any key)	Get ASYST'S 'OK' prompt.
STACK.DISPLAY	This command is optional. All it does is create a box on top of your display screen where ASYST will show you what is at the top of its number stack. This can give some insight into ASYST operation.
?FILE MYFILE	Opens the file and shows the four comments and file structure on the screen. Remember the sampling rate (shown in the fourth comment).
FILE>UNNAMED.ARRAY	Reads in first subfile to stack. This may occupy as much as half of ASYST's stack space. Therefore... stack space. Therefore...
CREATE.COPY MYARRAY	Get the data off the stack into a named array. We can use it repeatedly from there.
MYARRAY SUB[100 , 1024]	Get a 1024 long piece of MYARRAY (starting with the 100th value in MYARRAY) on the stack. At this point we could look at this data in the time domain simply by saying: DUP Y.AUTO.PLOT The DUP is necessary because Y.AUTO.PLOT deletes its input from the top of the stack.
LOAD FFT1024.LOG	Loads the "words" defined from a text file on your CATALYST SUPPLEMENTAL PROGRAM disk. These definitions need to be loaded only once per ASYST session.
200000000 FFTHANN	That's sampling rate in Hertz (from

the fourth comment in MYFILE), and the command to create a display of the FFT of the data after weighting the data by a raised cosine function to reduce leakage. The calculation takes about seven seconds.

At this point the stack is empty, and most of the screen is filled by the FFT results, as labeled. If your data has good signal-to-noise ratio, the bottommost labels on the Y axis may be zero. This is only because ASYST uses only three digits for labels; the values of the zero labels should be obvious from the sequence of the labels above. (Note: each factor of 10 in voltage is a 20 dB change in power.)

You may type STACK.DISPLAY to return the display to its previous status. You may get a different subarray of MYARRAY on the stack and play with it. You could read the next subfile from MYFILE and replace the data in MYARRAY. For details on these and other possibilities, please refer to ASYST's documentation.

B.2.6 Reconfiguring ASYST

Reading in 8192 real values requires ASYST to create a 32K byte transient array on its heap. Unlike earlier versions of ASYST, ASYST Version 1.5 cannot do this if exactly 32K bytes are allocated as heap space. The allocated heap space is easily changed with the RECONFIGURE command. Enter at least 33K, preferably more, as the heap size. If you don't have ASYST modules 3 or 4, you may enter 0 for the Data Acquisition and GPIB buffer size to free up as much memory as possible. You must save your reconfigured ASYST to disk; name it something other than ASYST so it does not replace your original version. Now, instead of typing ASYST to execute the original version, type the name of your reconfigured version to execute it. This will eliminate the "Error 81 - not enough space for transient array" message.

APPENDIX C
Function Generator Utility

FGEN is a utility program provided on the CATALYST Program Disk. It mathematically generates functions for use by the 8601 Quad Programmable Function Generator. It permits the creation of complicated functions without penalizing those users who need only a simple function. Several levels of complexity are discussed below.

Loading functions into the 8601's memories is controlled from CATALYST's 8601 setup menu. The functions do not have to be created by FGEN but can be files stored from CATALYST (See Section E.4).

C.1 One Function Per Channel

The easiest way to use FGEN is to associate one of the standard functions with each channel, tell FGEN you do not wish to define another set of functions to add to or multiply by this set, and then tell FGEN to replicate this set (which is 8192 words long) a sufficient number of times to fill the memories attached to the 8601.

When only one set of functions is defined the 'relative amplitude' and 'offset', which you are asked to specify, have no effect except that relative amplitude cannot be zero. The use of relative amplitude and offset is discussed in the next section. (Note: the actual amplitude and offset of the 8601's output is controllable from CATALYST's 8601 setup menu.)

The standard functions provided by FGEN are:

Sine:	Positive peak is +1, negative peak is -1. This is then multiplied by relative amplitude, then offset is added.
Triangle:	Peaks are +1 and -1. This is then multiplied by relative amplitude, then the offset is added. The waveform starts at the cell after a negative peak.
Pulse train:	'On' level is 1, 'Off' level is 0. FGEN asks for the duty cycle, which is the percent of each cycle the output is 'on'. The result is then multiplied by relative amplitude, and then offset is added. The 'On' period begins in the first cell.

Each of the above functions ask you for number of cycles desired in 8K words of memory. It is recommended that you specify as low a number as possible, such as 1, and then use a clock speed on the 8601 fast enough to get the frequency you want. The 8601's clock can be set as fast as 1 µsec per word.

If you specify a non-integer number of cycles there will be a

discontinuity between the last point of the 8K word block and the first point of its repetition.

FGEN also permits you to define a completely arbitrary function. You define a value at point 1, then select some point to the right and define its amplitude. The waveform is built by interpolating between the points you define. Any piecewise linear function (a sawtooth, bipolar pulses followed by zero, etc.) can be easily built this way. The accuracy with which nonlinear functions may be approximated depends only on how many points you enter.

If you select a point to define which is either already defined beyond the last point possible (these limits are displayed when you are prompted for the next point), the waveform is considered terminated at the last defined point (as explained in next section). If you define the last possible point, the waveform terminates at the last point.

C.2 Several Functions Per Channel

After you define one 'set' of functions (one for each active channel) you may define another set to be added to or multiplied by the result so far. This may be done as many times as desired. Relative amplitude is the amplitude of a signal in one set relative to the amplitude of a signal for the same channel in other sets. Offset is also useful in combining sets this way. Some examples should clarify how this is used:

	Function	Rel Amp	Offset	# cycle	
Set 1:	Sine	1	0	(a lot)	
* Set 2:	Sine	0.m	1	1	
Result: Carrier with m% amplitude modulation.					
Set 1:	Pulse train	1	0	100	50% Duty cycle
+ Set 2:	Pulse train	-1	1	10	5% Duty cycle
Result: Pulse train with every tenth pulse missing.					
Set 1:	Sine	1	0	(a lot)	
+ Set 2:	Pulse train	10	0	1	50% Duty cycle
Result: Waveform for checking for transient intermodulation distortion.					

It should be evident that simply adding two sine waves together makes a two-tone signal, and that multiplying a sine wave times a pulse train with 0 offset creates a sine burst.

It also is possible to intersperse arbitrary data with a standard function by: 1) defining the standard function; 2) multiplying by an arbitrary function which is all 1 where you want to keep the standard function, and all 0 where you want to replace it; 3) add an arbitrary function which is non-zero where you made the gap in step two.

Note that "relative amplitude" and "offset" specify only the amplitude and offset of a function relative to other functions which

will be added to it (or multiplied by it). The actual amplitude produced by the 8601 is controlled from WAVEFORM-CATALYST's 8601 setup menu.

C.3 Multiple Groups of Sets

Some applications may require an arbitrary waveform longer than 8K words per repetition. To do this, simply don't ask FEGN to replicate the 8K word result block (containing one or more sets of data combined together). FGEN will output the result block once, clear it, and then begin building the next result block by asking you to define its first set of functions.

It is also possible to make triangle or sine waves using two blocks per cycle. Simply create two successive blocks, each with 0.5 cycles, but give the first a relative amplitude of +1 and the second a relative amplitude of -1. (This technique is not extendible to four blocks per cycle because FGEN scales each block to completely use the 10 bit 'space' available in the 8601 D/A's. Therefore, the most positive and most negative values must appear in each block. For this reason the sine function is actually a cosine function.

C.4 Data Not Created by FGEN

Any time data stored in CATALYST's file format may be loaded into the 8601's memories. CATALYST will unpack byte data to 16-bit words for the 8601. The 8601 expects the least significant 10 bits (D0 to D9) of each word to be a data value, and the next two bits (D10 and D11) to contain a channel routing code. This is explained in the 8601's manual. Therefore, if data from an 8-bit transient recorder is sent to an 8601, all of its output will be on Channel 1 and can never exceed one fourth of the 8601's maximum output amplitude. If data from a 12-bit TR is used, any values exceeding one-fourth full scale will not be output by Channel 1, because the 8601 interprets D10 and D11 as routing information.

APPENDIX D

Sinusoidal Fitting

As an example of a USER written analysis processing routine, LeCroy includes "SINFIT", a sinusoidal fitting measurement routine based on the acquired data from the recorder. The test is activated by selecting "PROCESS=YES" in the digitizer setup menu (Figure 5.2). The conditions of the sine testing are discussed below. Selection of the input signal is important. For accurate results the input source should be of the lowest harmonic distortion. A low pass filter inserted at the output of the sine generator will remove higher harmonics that cause aliasing. Input amplitude of the sine wave is typically 50 or 80% of full scale of the recorder (e.g., 512 mV on TR8828C). A fitting is performed after each trigger. Arming the recorder with CATALYST single trigger followed by a manual trigger is is suggested. Waveform CATALYST will store the raw, unprocessed data. To exit from the process, press the space bar. Input signal frequency conditions are described below.

Testing of dynamic ADC performance is done by acquiring data from a sinusoidal signal source and comparing the reconstructed data to that of a theoretical ADC. A sinusoidal 4-parameter fit (amplitude, offset, frequency, and phase) to the acquired data is performed and the RMS error is compared to the pure quantizing error predicted for a perfect ADC.

The sinfit program requires that there are a minimum of 512 samples with at least 2 full cycles and at most, 32 cycles of the sinewave. This means that there are only limited ranges of input frequencies that can be used. For direct measurements:
.004 x sample freq. <= input freq. <=.06 x sample freq.

It is possible to fit higher frequency signals using the "beat" frequency generated by taking every nth sampled point. If there are more than 32 cycles, the program automatically tries to find the beat frequency by taking every 2nd, 4th, or 8th point and in this case up to 4k samples may be needed. At these high input frequencies the range depends on the beat harmonic as follows:

(1/n-.06) x sample freq. <= input freq. <= (1/n-.004) x sample freq.
or
(1/n+.004) x sample freq. <= input freq. <= (1/n+.06) x sample freq.

Sinewave amplitudes for these measurements are typically 50% (at high frequency) and 80% of the full scale range for the digitizer.

The sinfit procedure generates the following information:

A - amplitude of the signal

F - this is either the input frequency or it is the difference between the sampling frequency and 1/n multiple of the sampling frequency (when the input frequency gets close to 1/8 the sampling frequency)

Lost bits - This is an indication of the digitizer's error relative to an ideal one

Max error - maximum deviation from the ideal sinewave

S/N - signal-to-noise ratio

A plot is generated showing a bar graph to represent the deviations from the best fit sinewave, and the digitized sinewave with the same timebase.

For more information, see LeCroy Application Note 2004.

APPENDIX E: DIAGNOSTICS

If you experience a problem with your system, the following procedure is suggested to help you troubleshoot it.

1. Disconnect all peripherals from the computer.

2. Run the diagnostics provided with the PC. These are explained in detail in the "Guide to Operations" Manual".

3. If the computer passes all of the tests provided, run the GPIB diagnostics (IBDIAG) supplied by National Instruments. (See NI's manual for details).

4. Connect the LeCroy 8901A or 6010 Controller to the PC, turn on the crate, and run CATALYST. If you get a message "unrecognized crate controller" check the following:

   a) the LeCroy Controller is set for the correct GPIB address (remember, it cannot be address 0)
   b) two GPIB devices do not have the same address
   c) Cables - are they properly connected?
   d) GPIB ADDR on parameter table corresponds to controller GPIB address.
   e) Module Set Up - Slot # of module corresponds to number in parameter table; module is enabled for acquisition

5. If CATALYST was working fine but seems to have stopped:

   a) If you pressed SHIFT+PRTSC and a printer is not connected, or the printer is off line, you must connect printer and place on line.

      If printer not available, you must re-boot the system (CNTL+ALT+DEL).

   b) If the keys on the keypad don't seem to be responding, the NUM LOCK key might have accidentally been pressed. Press it and try again.

   c) Check the modules' setup to make sure they are properly configured.

SERVICE DOCUMENTATION

PRODUCT ASSISTANCE Answers to questions concerning installation, calibration, and use of LeCroy equipment are available form the Customer Services Department, 700 Chestnut Ridge Road, Chestnut Ridge, New York 10977-6499, (914) 578-6059, or your field service office.

WARRANTY LeCroy warrants its instrument products to operate within specifications under normal use and service for a period of one year from the date of shipment. Component products, replacement parts, and repairs are warranted for 90 days. Software is thoroughly tested, but is supplied "as is" with no warranty of any kind covering detailed performance. Accessory products not manufactured by LeCroy are covered by the original equipment manufacturers warranty only.

In exercising this warranty, LeCroy will repair or, at its option, replace any product returned to the Customer Service Department or an authorized service facility within the warranty period, provided the warrantor's examination discloses that the product is defective due to workmanship or materials and has not been caused by misuse, neglect, accident or abnormal conditions or operations.

The purchase is responsible for the transportation and insurance charges arising from the return of products to the servicing facility. LeCroy will return all in-warranty products with transportation prepaid.

This warranty is in lieu of all other warranties, express or implied, including but not limited to any implied warranty of merchantability, fitness, or adequacy for any particular

www.manualsplus.com
1033900

LeCroy